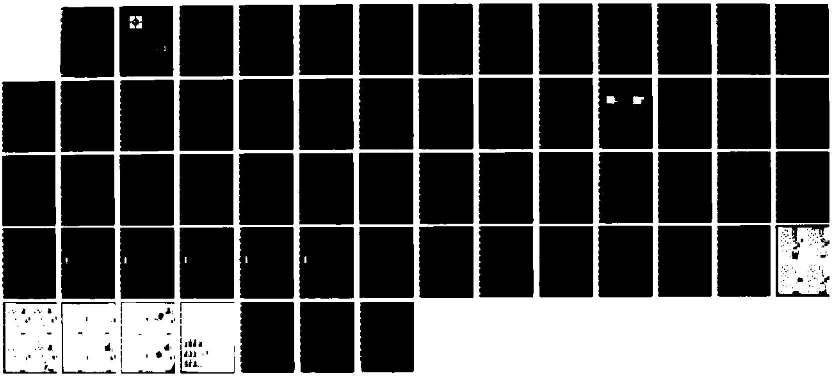
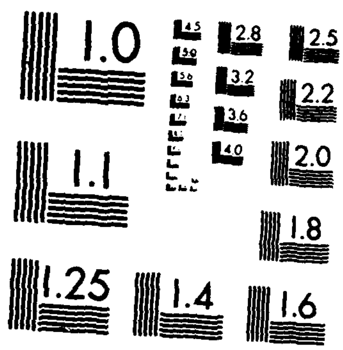
AD-A174 631 SATISFICING DECISION-MAKING IN SUPERVISORY CONTROL PART 2(U) MASSACHUSETTS INST OF TECH CAMBRIDGE MAN-MACHINE SYSTEMS LAB L CHARNY ET AL 31 JUL 86 N00014-83-K-0193 1/1

UNCLASSIFIED F/G 5/1 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

(12)

Man-Machine Systems Laboratory

Satisficing Decision-Making in Supervisory Control

Leonid Charny
Thomas B. Sheridan

July 1986

CONTRACT N00014-83-K-0193
WORK UNIT NR 196-179
ENGINEERING PSYCHOLOGY PROGRAMS
OFFICE OF NAVAL RESEARCH
ARLINGTON, VIRGINIA 22217

Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A174631

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED

1b. RESTRICTIVE MARKINGS:

2a. SECURITY CLASSIFICATION AUTHORITY:

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE:

3. DISTRIBUTION / AVAILABILITY OF REPORT: Approved for public release; distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S):

5. MONITORING ORGANIZATION REPORT NUMBER(S):

6a. NAME OF PERFORMING ORGANIZATION: MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6b. OFFICE SYMBOL (If applicable):

6c. ADDRESS (City, State, and ZIP Code): 77 Massachusetts Avenue, Cambridge, MA 02139

7a. NAME OF MONITORING ORGANIZATION: Engineering Psychology Program, Office of Naval Research (Code 442 EP)

7b. ADDRESS (City, State, and ZIP Code): 800 North Quincy Street, Arlington, VA 22217

8a. NAME OF FUNDING / SPONSORING ORGANIZATION: Office of Naval Research

8b. OFFICE SYMBOL (If applicable):

8c. ADDRESS (City, State, and ZIP Code): 800 North Quincy St., Arlington, VA 22217

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: N00014-83-K-0193

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO
	NR	196	179

11. TITLE (Include Security Classification): Satisficing Decision-Making in Supervisory Control

12. PERSONAL AUTHOR(S): Leonid Charny, Thomas B. Sheridan

13a. TYPE OF REPORT: Final (Part 2)

13b. TIME COVERED: FROM Mar 83 TO July 86

14. DATE OF REPORT (Year, Month, Day): 1986 July 31

15. PAGE COUNT: 91

16. SUPPLEMENTARY NOTATION:

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Supervisory control; mental models; decision aids; human-computer interaction.

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This paper describes a flexible graphics system GraMAD for aiding a human decision-maker in making a selection out of a discrete set of alternatives while trading off several criteria. Three major components of this selection process, called satisficing, are identified and three modes of information presentation to the decision-maker are studied. Necessary elements of multiple-objective computer aiding systems are discussed. Results of experiments with human subjects working with the GraMAD system are discussed.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL: Thomas B. Sheridan

22b. TELEPHONE (Include Area Code): 617-253-2228

22c. OFFICE SYMBOL: 3-346

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

MAN-MACHINE SYSTEMS LABORATORY
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge MA 02139

Satisficing Decision-Making in Supervisory Control

Leonid Charny
Thomas B. Sheridan

ABSTRACT

This paper describes a flexible graphics system GraMAD for aiding a human decision-maker in making a selection out of a discrete set of alternatives while trading off several criteria. Three major components of this selection process, called satisficing, are identified and three modes of information presentation to the decision-maker are studied. Necessary elements of multiple-objective computer aiding systems are discussed. Results of experiments with human subjects working with the GraMAD system are discussed.

1. Introduction: Conflicting Decisions Objectives in Supervisory Control

The problem of selecting a control action has a special importance for supervisory control. Considerable success has been achieved in modelling continous manual control systems [Sheridan and Ferrell, 1974]. It has turned out to be a much more difficult task to model human decision making which involves thinking and cognition. There has been an extensive debate whether it would be possible eventually to completely remove the human from a control loop of a process (as it has been the case in many manually controlled tasks). It may seem reasonable to assume that if we have an adequate model of human control actions then we can program this model into the computer. Then nothing would stop the computer from replacing the human. Actually this logic has been propelling a great deal of Artificial Intelligence research.

It is doubtless that as the technology progresses, more and more systems will become more human-independent. However, regardless of the technological advances, many systems will always require human presence in their control loop if for now other reason than to evaluate alternative situations. In the present research we focus on a particular class of these systems - ones which incorporate conflicting control objectives of their operation.

There are many reasons why subjective value judgements have to be made by operators controlling even physical, not to mention socio-economic systems. One of the reasons is that systems cannot be modelled sufficiently (for control purposes) accurately. In this case the value judgements work as heuristic rules for "trimming" control. Conflicting objectives, on the other hand, are present in almost every sufficiently complex system operated by man. Commonplace problems are to trade-off safety v.s. expenses, speed v.s. accuracy, immediate benefit v.s. long range gains - all of which are encountered in practice in a variety of tasks.

Using the utility theory as a basis for modelling supervisory control paradigms have been suggested in [Sheridan, 1976]. Real time subjective judgements by the operator controlling some system reflects our inability to build an adequate mathematical model describing the system. The utility function of the operator in the sense of [Sheridan, 1976] can be viewed as her/his mental model of the controlled process. Computer control as utility function maximization has been suggested in [Yufik, 1985].

Classical decision theory has very stringent conditions for the utility function to exist. In particular it requires one to consider a whole set of possible outcomes which the operator should take into account when making her/his decision. In this research we consider outcomes of the operator's actions being multi-attribute events with measurable attributes [Keeney and Raiffa, 1976].

In supervisory control the operator acts as a decision-maker (DM) performing discrete actions for controlling the system. Objectives the operator tries to optimize are frequently in conflict with each other, so that it is impossible to achieve optimal performance simultaneously and independently for each them.

For example, the operator needs to choose one out of two alternative actions A and B. The outcome of each action results in different values of operating pressure and temperature. The operator wants to maximize the pressure and minimize the temperature. Action A results in a pressure rise to 300 p.s.i. and a temperature rise to 245 F; while action B results in a pressure fall to 130 p.s.i. and a temperature rise to 400 F. The objectives to achieve simultaneously highest pressure and lowest temperature are mutually contradicting. The operator has to make a trade-off between preferences.

Potentially the computer can provide the operator assistance in scanning the full range of possible outcomes of her/his decision alternatives and making a control decision at each step. The computer can house a data base with a direct access to all measured plant parameters and can store heuristic information. The human - computer interaction and the nature of this computer aid are the factors of key interest to us.

2.1 Satisficing

Our interest is in helping the operator to decide what he wants in light of what is available. The task we are considering is a part of *planning* in terms of the general supervisory control paradigm [Sheridan, 1976]. In this paradigm the supervisory operator decides what s/he wants the process under her/his supervision to do, i.e. what subgoals s/he wants the automatic system to strive to achieve. This is a multiple-objective decision making problem where a decision consists of selecting and ordering decision elements or actions, out of a large set of available ones. The operator needs to find a satisfactory sequence or strategy (as compared to an "optimal" one, since no unique utility function is assumed) and since the operator has finite time and energy to consider the options. This problem has been called *satisficing* in the literature [Wierzbicki, 1982].

It is natural to consider selection of a sequence of actions as compared to a single action when supervisory control planning is concerned. From this the operator may actually initiate just one control action, which in turn would trigger a chain of simpler actions performed by the computer. A simple example of this is an act of selection of a multi-step control algorithm - the operator just selects or builds one algorithm, and the computer then goes through a whole series of steps. It seems natural to assume that the operator has some good idea as to what the consequences of one simple action would be. On the other hand s/he needs a help in evaluating cumulative consequences of a "chain reaction" triggered by her/his selection.

A process of selecting the "best" or satisficing action could be schematized in the following way:

1. All possible sequences of actions are somehow considered. The operator may be aware of all simple actions involved. But the number of possible sequences triggered by her/his control decision could be quite large and not easy to enumerate in her/his mind.
2. From the whole set of consequences the operator selects a few likely candidates to further consider as "the best" action. The operator may need to visualize in some way the whole set of *consequences of her/his alternative decisions.* Since the consequences are associated with a multi-dimensional space, it may not be an easy thing to do.
3. The operator revises the few selected alternatives and selects the "best" or "satisficing" one.

2.2 General Paradigm

We now specify the studied paradigm in exact terms. The DM must select a sequence of actions or strategy out of a large set of available ones. Each strategy $\underline{u}_i$ is comprised of a sequence of up to N primitive actions u_i^j performed in the order defined by this particular strategy:

$$\underline{u}_i = \{u_i^1, \ldots, u_i^N\}$$

Each primitive action u_i^j as well as transition $(u_i^j,\ u_k^l)$ from one action to another results in pay-offs and/or losses for the decision maker along some different attributes (M reward functions altogether) :

(Actions)

$$c_{1,i}^{\,j}(u_i^j)$$

$$c_{2,i}^{\,j}(u_i^j)$$

$$\ldots$$

$$c_{M,i}^{\,j}(u_i^j)$$

$$c_{1,i}^{\,j}(u_i^j,\ u_k^l)$$

$$c_{2,i}^{\,j}(u_i^j,\ u_k^l)$$

(Transitions when actions u_i^j and u_k^l could performed one after another)

$$\ldots$$

$$c_{M,i}^{j}(u_i^j,\ u_k^l)$$

Each strategy $\underline{u}_i$ as a whole therefore is rewarded with some functions of individual action rewards and possibly of the strategy itself:

$$\begin{aligned} c_1 &= F_1(c_{1,i}^{1},\ c_{1,i}^{2},\ldots,c_{1,i}^{N},\ u_i^1,\ u_i^2,\ldots,u_i^N) \\ c_2 &= F_2(c_{2,i}^{1},\ c_{2,i}^{2},\ldots,c_{2,i}^{N},\ u_i^1,\ u_i^2,\ldots,u_i^N) \qquad (2.1) \\ &\ \vdots \\ c_M &= F_M(c_{M,i}^{1},\ c_{M,i}^{2},\ldots,c_{M,i}^{N},\ u_i^1,\ u_i^2,\ldots,u_i^N) \end{aligned}$$

Problems of this sort could be easily visualized with a partially connected decision graph. Each simple action is associated with a particular node. Arcs of the graph connect nodes with all other nodes which correspond to possible next step actions. All M costs are associated with each node (action) and arc.

An example of a situation described by this paradigm is a selection of a flight route for a crop duster mission. The map of the flight region could be discretized and a simple decision u_i^j would be to fly over a quadrant (i,j) on the map, or go through a corresponding node on the graph. (The altitude is omitted in this example). The starting and terminating nodes could be the same, so that the aircraft will return to its base after having completed the mission. A strategy to be selected consists of visited quadrants. These quadrants cannot be visited in a random order, and a connectivity map is also specified. A total number of available strategies is determined by a total number of all possible round trip routes emanating from the starting node and satisfying certain constraints (maximal time and fuel, minimal number of fields dusted).

The following 4 attributes may be considered in this problem:

- $c_{1,i}^{j}(u_i^j)$ - potential flight danger in the quadrant u_i^j (trees etc.)
- $c_{2,i}^{j}(u_i^j)$ - expected value of dusting in quadrant u_i^j
- $c_{3,i}^{j}(u_i^j,u_k^l)$ - fuel expenditure for going from quadrant u_i^j to quadrant u_k^l.
- $c_{4,i}^{j}(u_i^j,u_k^l)$ - time for going from quadrant u_i^j to quadrant u_k^l.

All these attributes are assumed incommensurable and therefore constitute the trade-off decision to be made and to be aided. The planner (operator) actually has four possibly conflicting goals: to minimize values of the time, fuel and danger exposure; to maximize expected number of fields dusted.

A possible way to help the operator (decision-maker) is to determine her/his own utility function. Then this function could be encoded into the computer and the computer could find the strategy which maximizes this utility

function. Methods of conventional optimization could be applied for solving this problem.

A number of ways to obtain the utility function analytically are suggested in [Keeney and Raiffa, 1976]. Unfortunately none of the methods of classical decision theory seems fit for our purposes. There are several reasons for that. To name just a few:

- Obtaining a comprehensive utility function is a very tedious and time-consuming process. It is especially lengthy when the dimensionality of the attribute space is large. If the speed of the decision making is important then we have to use some other ways to help the operator.

- The utility function may not be adequately determined outside the narrow range of attribute values which are close to the ultimate maximum. The DM may not care too much what her/his preferences are for the alternatives strategies that result in "remote" consequences (not close to the ultimate optimum). On the other hand who could tell which alternative strategy is *close* or *far* from the optimum before the latter has not been found? What measure of "closeness" should one use in narrowing down the set of alternatives, so determining the utility function would be easier?

- Our initial paradigm assumes that the operator has to make decisions very often. Her/his system of preferences could drift with time as a means of adopting to the ever changing system's behavior. The operator's learning curve for controlling a complex process does not flatten out for a long time. Accordingly, the computer aid should be able to account for an apparent instability of the operator's utility function.

All things considered, it has been decided that the classical decision theory is more a provider of spiritual guidance than a practical tool for approaching the problem. For all practical purposes it is not tractable to determine one's multi-attribute utility function. A decision has to be made nevertheless. The problem therefore could be formulated as to how to help the operator come satisfactorily close to maximizing her/his utility function without explicitly determining that function.

2.3 Multiple Attribute Optimization Problem

In this research the problem of finding the "best" strategy is treated as a discrete multiple-attribute optimization problem (MAOP). Each alternative strategy is represented as a point in an M-dimensional space of attributes a_i $(i=1,\ldots,M)$. The attribute values of each decision strategy are calculated using expressions (2.1). Accumulated reward/loss functions (2.1) are assumed to be sums of rewards/losses in each node of the decision path. This

assumption considerably limits the range of applications of this computer aid. However our main purpose of the aid was not to solve actual applied problems, but to study what interface features are useful for the human decision-maker.

The DM wishes to find a point which has maximal possible values of coordinates along some axes (*positive* attributes) and minimal possible values along others axes (*negative* attributes). (Whether DM's wish is to maximize or minimize the value of a particular attribute depends on the nature of these attributes).

In other words, the discrete problem considered in this research is:

Find vector $\underline{u} = (u_1, u_2, \ldots, u_M)$ from a finite set of N vectors $\underline{u}_i = (u_{i,1}, u_{i,2}, \ldots, u_{i,M})$ $(i=1,\ldots,N)$, such that it has MAXIMAL value of ALL coordinates $j : (1 \leq j \leq P \leq N)$ and MINIMAL value of ALL coordinates $k : (P < k \leq N)$.

The are two specific assumptions of this approach:

1) It is assumed that <u>any</u> decision maker solving this problem wants to either maximize or minimize values of the attributes. It is different from the case when the problem requires the DM to get a value close to some given reference value. This effectively means that utility function of any attribute is a *monotonic* function of the values of the attribute (either non-increasing or non-decreasing). The attributes with a monotonically non-decreasing utility function are called *rewards*, and the attributes with a monotonically non-increasing utility function are called *losses*.
2) The objectives are assumed to be <u>preferentially independent</u>, i.e. regardless of the values of some attributes, the DM still wishes to maximize/minimize values of the other attributes.

This problem has been extensively researched. However, most attention has been paid to a continuous formulation of a similar problem:

Find $\underline{u} = (u_1, u_2, \ldots, u_N) \in U$, which delivers a maximum to ALL functions f_i of $\underline{u}$:

$$f_1\ (u_1, u_2, \ldots, u_N) \dashrightarrow MAX$$
$$f_2\ (u_1, u_2, \ldots, u_N) \dashrightarrow MAX$$
$$\ldots$$
$$f_M\ (u_1, u_2, \ldots, u_N) \dashrightarrow MAX$$

Unfortunately these statements (as well as any other statement attempting to formulate MAOPs in purely mathematical terms) lacks meaning because there is no ordering on the set of all vectors $\underline{u}_i$ (or vectors $\underline{f} = (f_1, f_2, \ldots, f_M)$ in the continuous formulation). Only a notion of non inferior, or Pareto optimal, or efficient set of alternatives allows precise mathematical definition. This notion plays a crucial role in the multi-objective optimization.

Intuitively a Pareto set of alternatives is a subset of the set of all alternatives which cannot be obviously improved. In other words, there are two alternative vectors:

$\underline{u}_i = (u_{i,1}, u_{i,2}, \ldots, u_{i,M})$ and $\underline{u}_j = (u_{j,1}, u_{j,2}, \ldots, u_{j,M})$ such that:

$u_{i,k} > u_{j,k}$ for any *positive* attribute k and

$u_{i,l} < u_{j,l}$ for any *negative* attribute l,

while for at least one attribute either of the above inequalities become a rigorous inequality. Then it is obvious that for any DM alternative i is preferable to j, or j is dominated by i. A Pareto set is composed of all those alternatives which have no dominants. It is obvious that the ultimate "optimal" final solution has to belong to the Pareto set.

Since MAOPs are not amenable to cut-and-dried formulations (and therefore solutions), a variety of approaches have been reported in the literature. Several reviews and comprehensive books on the subject of multi-objective optimization techniques are available (e.g.[Cohon, 1978; Rosenthal, 1985]). Two major questions emerge however, regardless of one's approach to the problem:

1) How to generate a Pareto optimal (or efficient) set of alternatives?

2) After the Pareto set has been generated, how to select the "best" or the "optimal" final solution by the DM's own criteria?

The first question poses a purely mathematical problem. This problem could be unambiguously stated in precise and objective terms and is independent of a human DM. In other words, to solve this problem one does not need to know what preferences the human DM has. A number of algorithms have been developed for this purpose. This problem is not addressed in the present research.

The second question is markedly different. Suppose that, using one of the algorithms mentioned above, we have generated a Pareto set. Suppose further that this set contains 100 alternatives. What should we do next? How could we show this Pareto set to the human operator, our decision-maker?

This problem is ' . one we intended to study in this research. We will argue that this is purely a person-computer interaction problem, which cannot be resolved without close cooperation with the actual decision-maker. In our experiments we always used a Pareto set of alternative as an actual set to select from.

We consider a human decision-maker solving a problem of selecting an alternative from a Pareto set. We propose that three independent entities involved in this problem should be identified to help define the MAOP:

1) The actual task which requires multi-objective formulation. That is: what physical (or *actual*) problem does one try to solve using a multi-objective representation? Knowing the attributes and the set of alternatives to choose from is not enough to make a

selection. Suppose for example, that there are two attributes, speed and accuracy, and one is selecting a robot manipulator based on these two parameters. A set of alternatives could be taken from Fig.1, Prob.1. However it matters a great deal (for the final choice) what one intends to do with this manipulator. Depending on whether one needs the manipulator for operation in a dangerous environment, or for moving bulky items in a warehouse, one would pick a different alternative.

2) The topology of the Pareto set (i.e. the properties of the set which do not depend on the measuring scales of the attributes, e.g. number of points having the same value of a particular attribute). For different problems involving the same set of attributes (e.g. Speed and Accuracy) the same Pareto set of alternative could be generated. Even if two physical problems are fully different in nature, but have the same number of attributes, their Pareto sets could look identical if appropriate linear transformation of the attributes is selected (Fig. 1). Topology of the set however plays an significant role in selecting a point from that set. For instance, points could form clusters and some techniques based solely on the topology of the set could be used for reducing it's size (see for instance [Rosenman and Gero, 1985]). Obviously, reduction of the Pareto set simplifies a problem of presenting it to the human DM.

3) The human decision-maker.
It is evident that two different human decision-makers, while solving the same practical problem and facing the same Pareto set, will be likely to select different alternatives. This consideration effectively renders useless any way of solving the MAOP which excludes the human.

Classical utility theory concentrates on 1) and 3). It effectively states that it is enough to establish preferences of a particular DM, who solves a particular problem. If these preferences have been established for the whole range of decision attributes, then it is a trivial matter to apply them to the actual set of alternatives. Therefore the classical utility theory completely neglects any effect the topology of the set of decision alternatives may have on the DM. In other words, it is assumed that the DM preferences do not change after having seen *what is available* (rather than knowing *what may be available*). The best treatment of this approach could be found in [Keeney and Raiffa, 1976].

A number of other approaches in some way undermine the importance of the human decision-maker. Some of these, the so-called Generating Techniques [Cohon, 1978], have as a goal to solicit as little information as possible from the DM and then do the number crunching in the computer. The usefulness of these techniques seems very questionable. One approach which has gained great

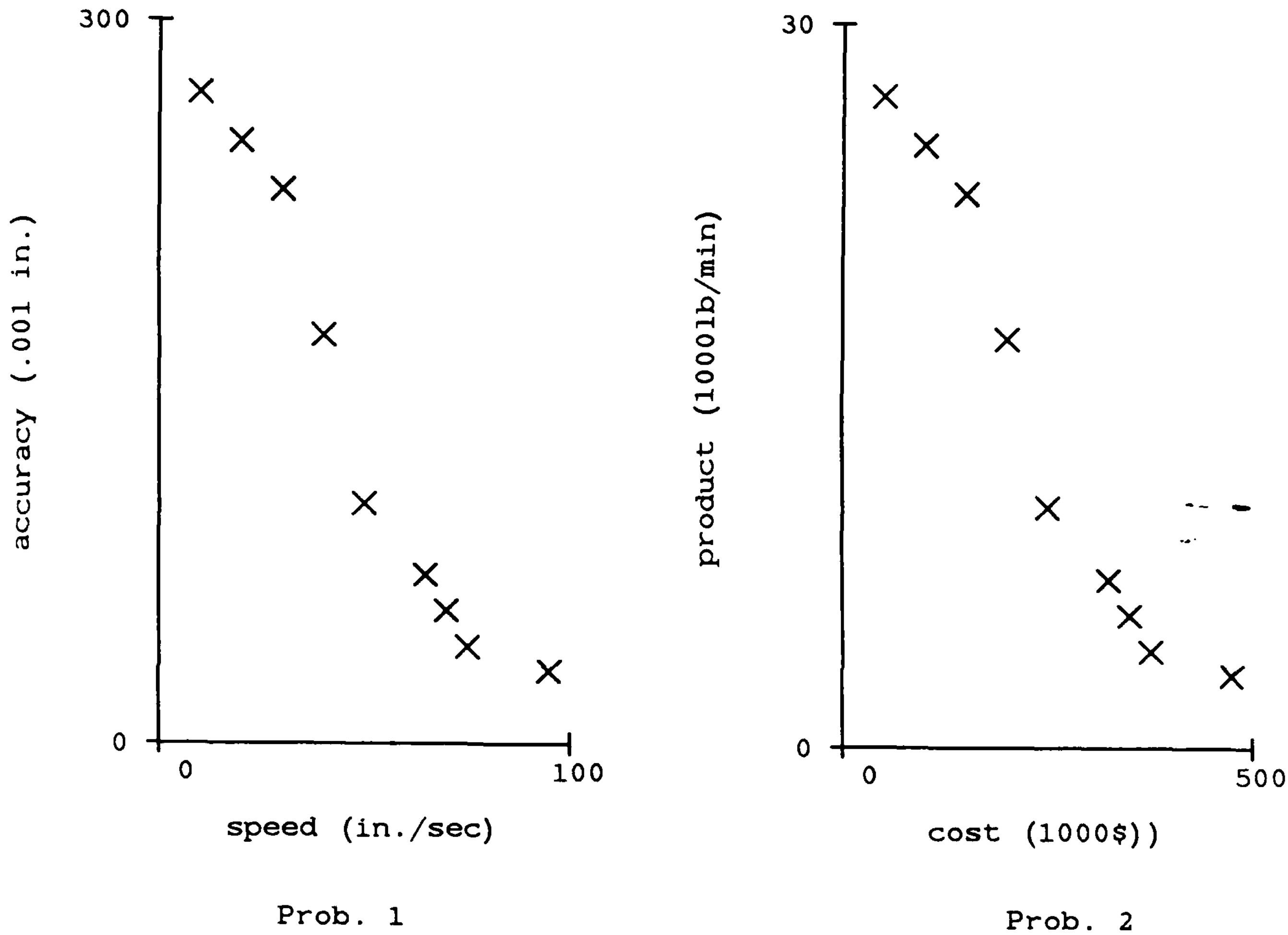


Fig. 1

Two different problems having

similar Pareto optimal sets

interest in recent years assumes that the DM not only has a global utility function that is meaningful but also has no felt need to converge on a single optimal point - there are regions in attribute space within which further search is not worth the effort.

It is our conviction that the human-computer interaction is a pivot for solving any MAOP. No one but the DM could assuredly exclude any alternative from the Pareto set. It is also our assertion that the DM's system of preferences is not stationary. It is shaped during the process of solving the problem and during the DM interaction with the Pareto set of available alternatives, and this is part of "satisficing". The proposed computer aid should account for this.

3.1 Design of a DM Computer Aid : General Requirements

Based on our view that human-computer interaction is the core of the solution of MAOP and that explicit utility function methods are unsatisfactory, we have developed an experimental computer decision aid, GraMAD (Graphic Multiple Attribute Decisions) to study how a DM might solve a MAOP in the most convenient way. This research also intended to highlight experimental observation of human subjects using the GraMAD. It has been found that systematic experiments have seldom been reported in the literature for human subjects interacting with a sufficiently complex computer aid.

A number of computer aids reported in the literature were designed for solving MAOP for particular applications. In this research we created a flexible graphical computer aid, which could be used for solving any discrete MAOP. It seemed that graphical human-computer interaction allows flexibility and is also more convenient for human decision-makers to use (as compared to batch or purely lexical dialogs). Some studies of graphical multi-attribute information presentation have been previously done in the Man-Machine Systems Laboratory [Barrett, 1980 and Knepp, 1981]. However these studies did not address a problem of satisficing or MAOP.

We intended to study which of the three stages of selecting the "best" or "satisficing" action is more difficult and/or more important for the human decision-maker, and where aiding might help the most. These decision-making stages as mentioned above (ch. 2.1) are:

1) enumeration of all possible decision sequences;
2) pre-screening and visualization of the whole set of consequences of alternatives decisions (visualization of the Pareto set);
3) final selection of the best decision out a few pre-selected ones.

These stages were reflected in three alternative forms of human-computer interaction and information presentation to the human subjects tested in this study. Each of the three information presentation forms focused on one of the three stages. We call these forms: "Full Graph", "Complete Set of

Cumulative Totals" and "Record of Best Cumulative Totals". They will be described in the next chapter.

3.2 GraMAD : a system for Graphical Multiple Attribute Decision

The GraMAD created in the course of this research can handle any number of attributes and any decision graph. It is not geared to any specific application. The human-computer dialog provided by our computer aid has several major options common to all three representations.

The DM arrives at the "close enough to best" or satisficing solution after going through a number of iterations. At each iteration the DM asks the computer to solve some single-objective optimization problem. The computer-generated solution of this intermediate problem is used by the DM for specifying conditions for the next iteration. At some point the DM decides that a satisficing solution has been found.

The following two requirements for any interactive decision aid for MAOP were considered necessary:

1) Capability of scanning the full Pareto optimal set of the problem. This means that the computer aid should be able to pick any alternative point from the Pareto set, if the DM wishes to. It reflects the fact that nobody but the DM her/himself can reject any point of the Pareto set.

2) Capability to react to "natural" articulation of preferences by the DM. This means that at any iteration the DM should be able to specify the condition for the next iteration in a natural way. What is "natural" will be discussed in section 5 where experimental results are reported.

It was thought that both of these requirements would be satisfied if an interactive combination of two simple and popular techniques - the weighting and the constraint methods - were employed.

At any iteration the DM would specify to the computer weights w_i and constraints C_i on values of the attributes. These weights would then be used by the computer to generate a linear objective function:

$$F = \sum_{i=1}^{M} w_i a_i \qquad (3.1)$$

where $w_i > 0$ is a weight of the i-th attribute and a_i is the value of the i-th attribute if the attribute is positive (a_i is minus the value of the i-th attribute if the latter is negative). It is required that the weights sum to 1:

$$\sum_{i=1}^{M} w_i = 1$$

The computer then maximizes this function F subject to the constraints on values of the attributes, also specified by the DM at this iteration:

$$\text{(3.2)} \qquad \begin{array}{l} \text{MAX } F\text{, subject to} \\ a_i > C_i\text{, for positive attributes; and} \\ a_i < C_i\text{, for negative attributes.} \end{array}$$

Note that the constraints effectively specify lower limits of values of positive attributes and upper limits of values of negative attributes.

It follows from the Soland theorem [Soland, 1979] that any point of the Pareto optimal set could be obtained as a solution of this optimization problem, assuming proper weights and constraints have been selected. This effectively proves that the first of the stated requirements is satisfied. In chapter 5 we discuss how natural it is for the human DM to specify weights and constraints.

It is noteworthy that the weights that DM subjects use play a very important role in the psychological literature. However, weights alone are not sufficient for selecting all points from the Pareto set. In other words, knowing the weights for all attributes is not enough for specifying the subjects' preferences. For instance, the alternative marked "o" will never be picked up by the computer if only weights are used by the DM for specifying preferences (Fig.2), since weights merely specify a direction in the attribute space.

On the other hand, appropriate selection of constraints alone is sufficient for determining any point of the Pareto set. In particular if a point's attribute values are specified as constraints (maximum on undesirable attributes, minimum on desirable), then optimization (3.2) with any objective function (3.1) yields this point. (This follows from the definition of the Pareto optimal set).

The DM can specify and independently alter both the weights placed on the values of each attribute and the constraints on the total cumulative values of each attribute. The computer then may suggest a sequence of decision elements (path on the decision graph) based on these specifications. The computer in this case displays the total cumulative values of the attributes for this path in bar-graph form.

To specify both weights and constraints is redundant. However it allows a human DM to express her/his preferences in a natural way (discussed in chapter 5).

The following features are present in each of the three types of information presentation. The reader may cross reference Appendix C to see examples and pictures of the actual computer displays for each type, as well as the instruction for using the GraMAD.

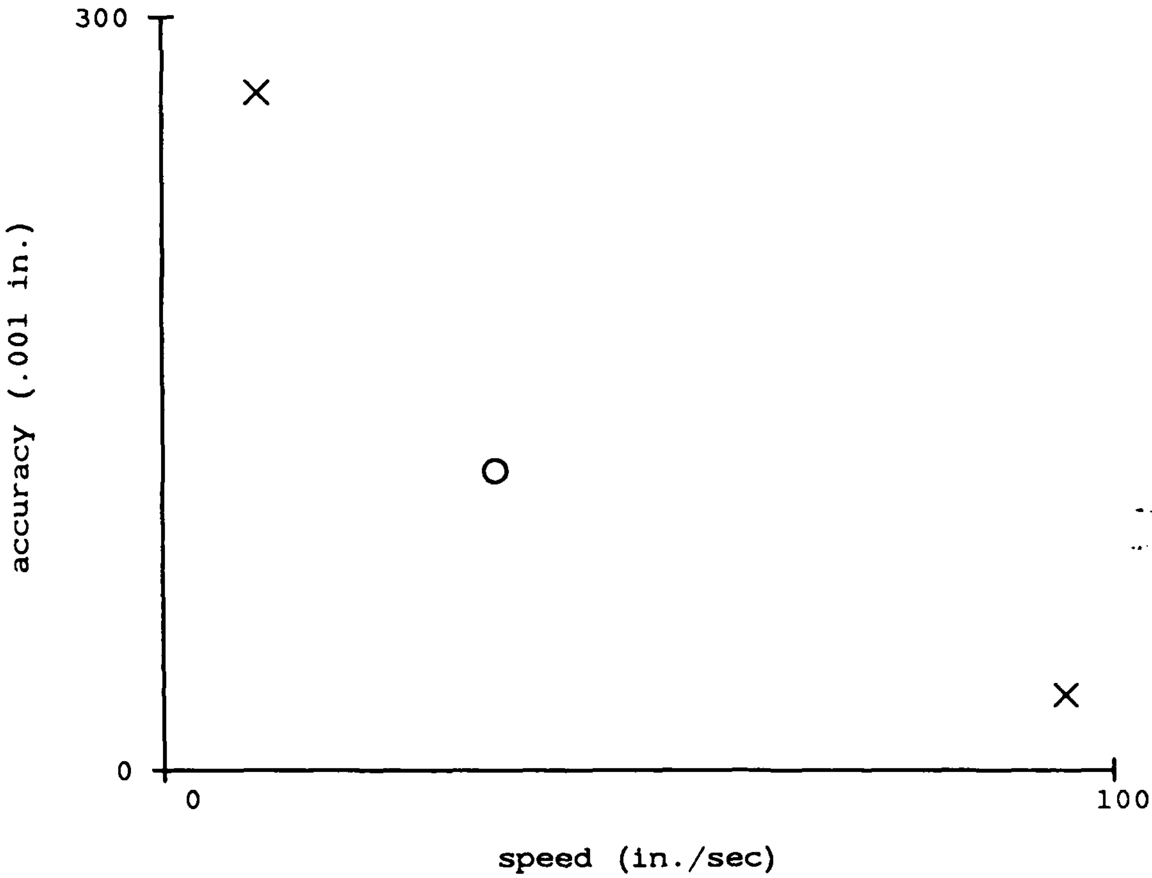


Fig. 2

Alternative marked "o" can never

be selected with linear weights alone.

1) Infeasible Constraints.

The DM may specify an infeasible set of constraints. This means that no available decision alternative could satisfy these constraints. In this case the constraints are treated as specification of an aspiration point (aspiration levels) in the attribute space (terminology used in the satisficing decision making theory [Wierzbicki, 1982]). The computer then tries to find an alternative which is closest to the aspiration point.

The question arises: how to measure closeness between points in the attribute space? Actually it is the basic question of multi-attribute optimization. Unless a utility function is known, no measure of closeness exists. However, as it has been noted above, appropriate selection of the constraints alone is sufficient for obtaining (by solving optimization 3.2) any point of the Pareto optimal set. Regardless what objective function is used, one still can scan the full Pareto set. This is one reason behind the approach used in [Wierzbicki, 1982]. In that work the problem of selecting a measure of closeness in attribute space is considered to be secondary to specification of the aspiration point.

In our experiments we studied the validity of the above consideration. An option to use the conventional satisficing approach has been implemented in GraMAD. In this mode DM can only specify an aspiration point and can not be able to affect the weights. The optimization 3.2 is then performed by the computer with pre-fixed values of the weights.

2) Sensitivity Analysis.

The DM is provided with an option to request a sensitivity analysis. In the context of this paradigm it means that s/he may ask the computer to show what minimal changes to the specifications (weights and constraints) are required in order to get a different recommendation from the computer. The computer actually displays for each weight what minimal and maximal values of this weight are, such that any change of this weight alone within the range of these values would not alter the computer-generated solution of the optimization 3.2. Mathematical details are explained in Appendix D.

When constraints specified by the human are infeasible, the computer shows how much each constraint should be relaxed in order that a feasible alternative can be found. Obviously this information would be displayed only for those attributes to which it applies - it is not always possible to relax just one constraint in order to make the set of constraints feasible.

There are broad implications of using sensitivity analysis (in this form) in computer decision aiding systems. DMs do not easily comprehand what the weights actually mean and what values should be assigned to them. Instead of blindly changing available parameters from one iteration to another, the DM is provided guidelines as to what the minimum changes should be.

The DM interacts with the computer using a mouse connected to a display cursor. The attributes are color coded. Information related to each attribute is displayed using the color assigned to the attribute. Change of weights and constraints is achieved by simple motions of the cursor on the display. The DM does not need to type in any numbers and/or words. The full capabilities of the GraMAD can be seen from the instructions on how to use it (Appendix C).

The following capabilities are provided for specific information presentation modes:

FULL GRAPH. The DM is presented with a computer-generated decision graph (see ch. 2.2). Colored bars (with colors corresponding to the attribute colors) poking from each node graphically display the amount of the attribute acquired at each node. The brightness of each node changes if the attribute weights are changed. A node's brightness reflects the node's "worth" in the case, assuming that the utility function is a weighted sum of the node's attribute values. This is the only time when implications of a linear utility function are explicitly displayed to the human.

The DM has the ability to select any number of nodes as the ones to be visited necessarily (mandatorily). The computer accounts for such selections in its advice. The computer also displays it's recommendations as paths on the graph.

We believe that visualization of the decision graph makes enumeration of the alternatives easy. This visualization also helps the decision-maker to get a better feeling of the problem. By selecting certain nodes, the DM is able to significantly reduce the space of alternative decisions (since there are just a few decision paths which go through any particular node(s)).

COMPLETE SET OF CUMULATIVE TOTALS. The purpose of this presentation is to help the DM visualize the alternatives geometrically as points in the attribute space. Here the DM has no way of controlling individual decision nodes. Each decision path is presented as a single point in the attribute space. Since the spaces beyond 3-D cannot be visualized by humans, the computer generates all possible combinations of 3 attributes and presents them as 3-D images on the screen. (A curious discussion on the human's ability to visualize spaces of dimensionality greater than 3 can be found in [Berkovich, 1977].)

Each 3-D projection of the attribute space is displayed as a cube with points inside it. The number of these points (the number of strategies) is equal to the total number of possible (distinct) routes connecting the starting and the destination nodes on the decision graph within the Pareto set. Each point inside the cube

represents a particular strategy (or alternative). The coordinates of the points inside the cubes are proportional to the values of the corresponding attributes.

Each cube can be rotated and magnified by the DM. The constraints can be visualized as colored parallelepipeds inside the cubes. Computer recommendation at each iteration is shown with a flashing point (one of the points inside each cube). The DM can specify with the cursor any point inside the cube and the values of it's attributes are then displayed on the screen. This point also gets marked out in all other 3-attribute cubes.

We believe that there has been little or no use thus far of such 3-D displays in multi-attribute decision making. It is presumed that in this representation, the DM can really visualize the set of available alternative decisions as a whole. Therefore one of our goals was to test whether a dynamic 3-D display can assist the operator working with computer decision aiding systems, and the degree to which he can model his own tradeoff or satisficing problem in this way.

<u>RECORD OF BEST CUMULATIVE TOTALS.</u> The purpose of this presentation is to give the DM a way to save some intermediate results obtained during her/his interaction with the computer aid. The saved results could be scrutinized later and the "best" solution selected out of them. These intermediate results are saved on the display in graphical form. Only what is selected by the DM gets saved. The DM can also restore any of the saved alternatives when s/he chooses to do so. Each alternative get restored along with weights and constraints which were used to generate it. The DM can effectively restart the iterations. In this mode no graphical representation of the problem is given: neither full graph nor the set of strategy points in the attribute space.

Of course, all of the above representations could be combined into a single display and/or integrated system. However our purpose was to study which of these representation is more important to the DM, and correspondingly, which of the phases of the selection process needs more aiding.

4.1 Experimental Evaluation of GraMAD

Experimental evaluation of computer aids for subjective decision-making is not simple. It is very difficult to evaluate something what does not have an objective measure. Unlike evaluation of an iterative computer algorithm, easily measured characteristics such as accuracy, number of iterations before conversion, and computational time, are not so relevant. "Accuracy" of the final choice does not have too much meaning, because no one knows what to compare it with. A number of iterations or interaction time are also not very good indicators of the aid's computational efficiency. On the contrary, the DM may be so displeased with the aid that s/he would try to shorten the interaction time and reduce the number of questions s/he poses to the computer.

One important consideration is whether the DM really cares or has anything like established preferences for alternatives in a test problem, because if not, her/his choices will vary randomly.

No reports of such experimentation (with an aid designed for a variety of applications) were found in the literature. The most common approach to testing flexible computer aids was to model the DM with a computer program-as-if this DM had a mathematically explicit utility function (e.g. [Marcotte and Soland, 1986]).

4.2 Experimental Design

A number of formal and informal experiments were conducted with GraMAD. In this report we discuss formal experiments with two subjects. At this stage of experimental development, much as in building expert systems, working intensively with a small number of "subject matter experts" was deemed to be more useful than amassing data across many DMs.

Experimentation in this area is very time-consuming. In our case the subjects had to spend time learning how to use the GraMAD faciley, and time was also required for subjects to get a "feeling" of the problem they were about to solve.

The experimental design had 6 problems to be solved by each subject. The problem contexts were completely unrelated with each other. Two problems were presented in a Full Graph form, two - in a Cumulative Totals form, and two - in a Record Best form.

Three of the problems were intentionally such that the human decision-maker might realistically be affected by her/his choice; these three "real" problems were presented to the subject in all three different forms. The description of the problems as they were given to the subjects can be found in Appendix A.

All problems given to the DM subjects had the same number of attributes: 5. However the number of positive and negative attributes varied from problem to problem. The number five was selected because the 5-D attribute space was sufficiently complex that it could not be visualized at once. At the same time it was a manageable size for sufficiently fast computation. The sizes of the Pareto sets were on the order of one hundred points.

An interactive computer program for building decision graphs was created which allows one to interactively build a decision graph of any shape and having any number of attributes. The values of the attributes for each node were assigned by the program using a random number generator; however these data could easily be altered by editing a single input. The graphs that were used for the problems are shown in Appendix A.

Before actually using the GraMAD, each subject was trained to develop a feeling for the problem and an idea about her/his preferences. The training was done automatically by computer. The subjects were given a set of pairs of alternatives, presented in the same graphical form to be used in the computer aid. The subject would select which alternative out of the two displayed s/he preferred, then articulate her/his preference by pointing with the mouse at one of three blocks "Prefer Left", "Prefer Right", "Indifferent" (Fig. 3). The attribute values of alternatives displayed at the training stage were randomly generated within the range of the attribute values of the actual problem. After each training session with each problem the recorded part of the experiment began. Subjects were asked to think aloud as they performed the tasks, and these comments were recorded.

A recording was done any time the subject requested computer advice after having specified her/his weights and constraints. At that point the specified weights and constraints were recorded along with the computer generated results of the optimization problem (3.2). The timing of such DM requests was also recorded by the computer. The recorded data are plotted for each problem/subject combination in Appendix B.

In the case of the Full Graph representation, subject DMs were asked to guess their best decision paths before using the aid. They were asked this question twice. At first they were asked to show the "best" path based only on the visual appearance of the decision graph. In this case the node brightness capability was turned off. The only clue the subject could use was the color bars reflecting amounts of attributes in each node. In conjunction with the second guessing the subject was able to use the brightness option and adjust the weights to his/her liking. However, no other capability was available beyond that. These guess solutions could be compared with the decisions obtained after an interaction with the full-capability GraMAD.

The time for solving each problem problem was not limited. The experiment was run by a computer program which presented subjects with a new problem after they were through with the previous one. The presentation forms were given in this order: Full Graph, Cumul. Totals, Record Best, Full Graph, Cumul. Totals, Record Best.

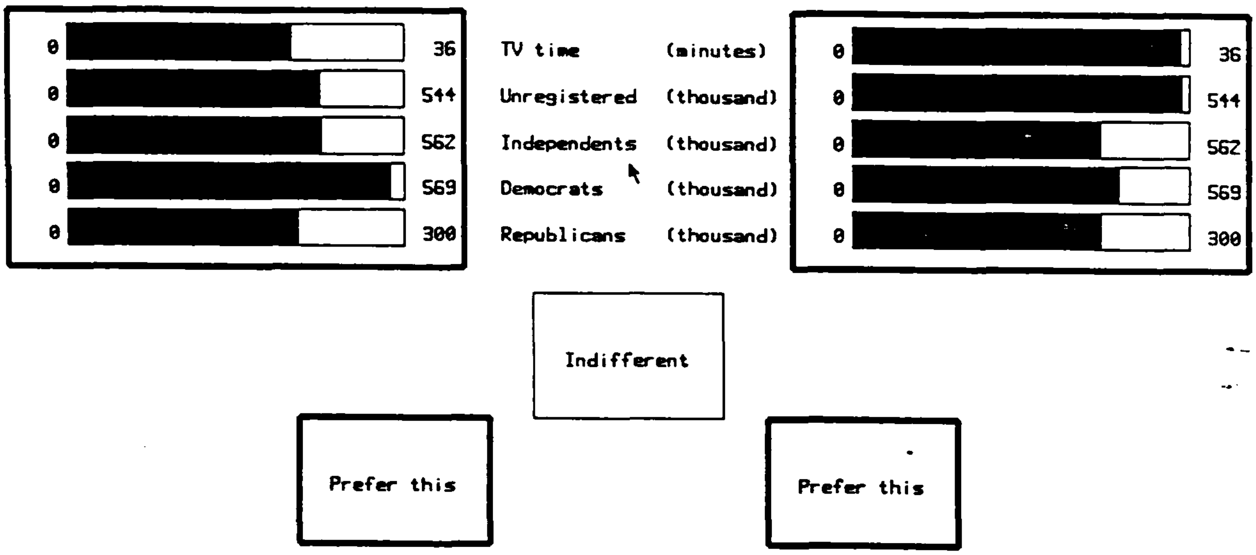

0 36 TV time (minutes) 0 36
0 544 Unregistered (thousand) 0 544
0 562 Independents (thousand) 0 562
0 569 Democrats (thousand) 0 569
0 300 Republicans (thousand) 0 300
Indifferent
Prefer this
Prefer this

Fig. 3

At the end of each experimental session the subjects were asked questions as to what features of the system they found to be the most/least useful. The subjects were also asked to suggest what capabilities it would be useful to add to GraMAD. Their comments and recommendations are discussed in Ch. 5. Some of the subjects' recommendations have been incorporated in GraMAD.

5. Experimental Results.

The most important information was recorded when a subject invoked the "Best Path" option. Each such invocation was an iteration in an iterative man-machine search procedure. Plots of the computer recorded data for all subject/problem combinations can be found in Appendix B. There are three types of plots corresponding to each subject/problem combination:

1. **Weight values** (Fig. 4). Values of the weights assigned by the subject are plotted for each of the 5 attributes. These values are recorded as functions of iteration number. Dashed lines show the sensitivity range for each attribute's weight value at each iteration.

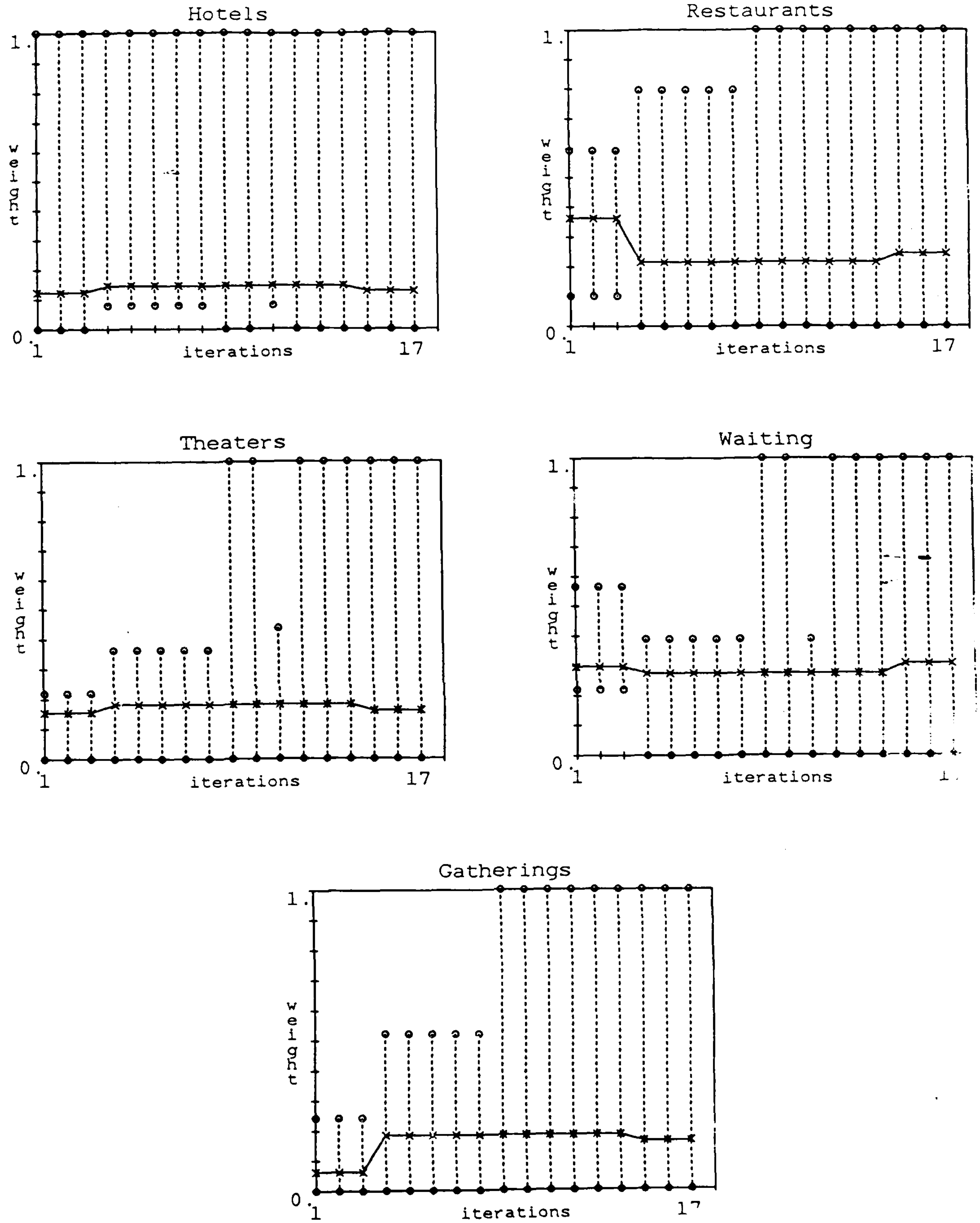
2. **Constraint/attribute values** (Fig. 5). Lower and upper constraints and computer-generated (after invoking the "Best Path") attribute values are plotted for each attribute as functions of iteration number. Lower/upper values are marked with little circles. The values of the attributes which are marked with "x" correspond to infeasible constraints.

3. **Time / feasibility** (Fig. 6). One plot shows amount of time (in seconds) the subject spent between each two consecutive iterations. The other plot shows a number of feasible alternatives at each iteration. For the "Totals" displays a third plot shows in which mode was the "Best Path" option invoked: whether the subject viewed all cubes or s/he was in a magnified mode. A piecewise-constant line is a function of time showing the mode: lower value corresponds to "view all cubes", upper value corresponds to a magnified cube mode. The "x" marks show when the "Best Path" option was invoked.

A very important conclusion has been made after studying these plots. It appears that it is impossible to specify a simple "convergence" criterion, i.e. when the DM decides that it is "enough" and the satisficing solution has been found. This observation once again emphasizes the difficulty of modelling the human part of this DM-computer satisficing dialog.

Another, in fact a surprising, result is that the "reality" of the problem seemed to have little to do with the subject's motivation to solve that problem (contrary to expected). The three "real" problems used in the experiments turned out to be the least exciting ones for the subjects. For example, in our problem where reward / loss were actual nickels (we had to constrain the magnitudes of the rewards) the subjects did not really care whether they would receive 10 or 25 nickels. The subjects seemed more preoccupied with minimizing the losses rather than acquiring the gains. Each of these problem had a "quits" alternative - such that one did not incur any losses and did not gain anything (the value of all attributes was 0). This alternative was actually selected by subject #2 while solving Problem #4.

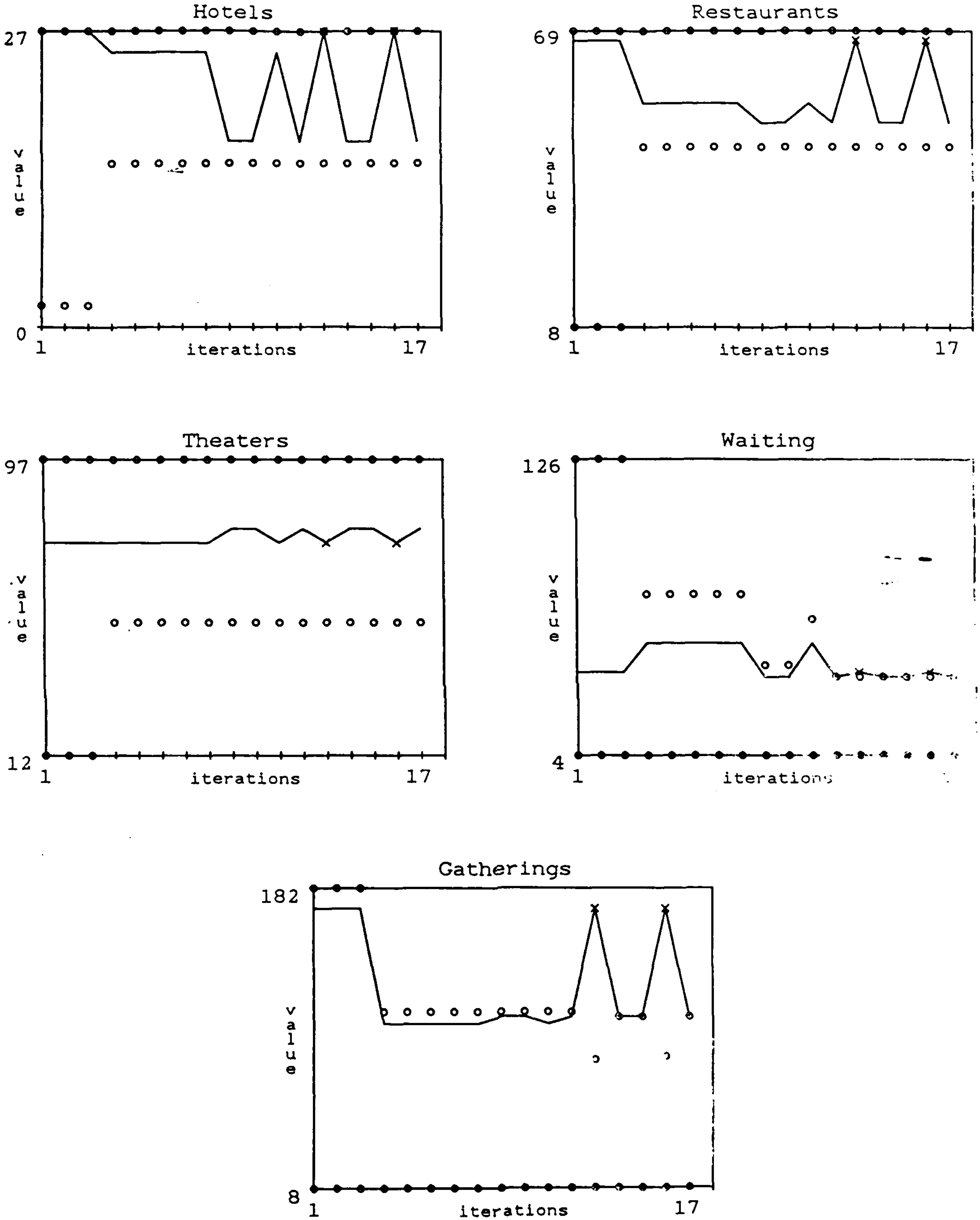
It turned out that the motivation to work on a problem was related to how interesting the problem seemed to the subjects, not to how real the problem



Full Map Display. #1 : "THE MARVELL TRIP "

Subject #1

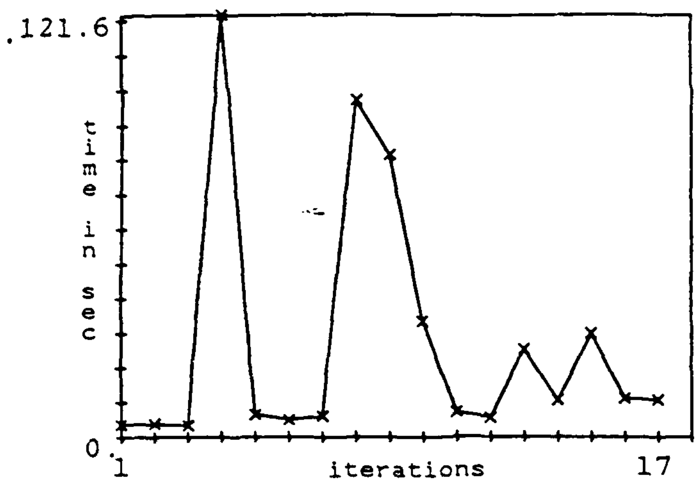
Fig. 4

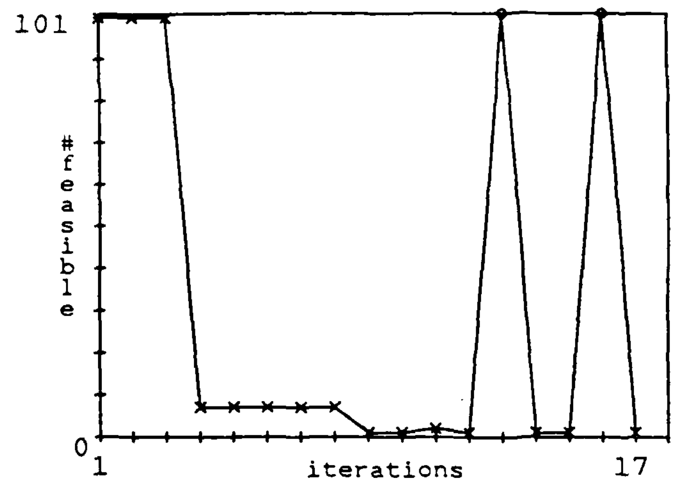


Full Map Display. #1 : "THE MARVELL TRIP "

Subject #1

Fig. 5





Full Map Display. #1 : "THE MARVELL TRIP "

Subject #1

Fig. 6

was. Somewhat not actually "realistic" problem of planning a political campaign tour (Problem #2) was found quite interesting by one of the subjects. Both subjects also got quite interested in Problem #1, "The Marvell Trip". The subject's interest in a problem was found to be reflected in a number of iterations the subject needed before arriving at the best solution.

It was observed that out of the three types of information presentation displays, only two were found useful by the subjects: Cumulative Totals and Record Best displays. Full map display was found to be of no use by both subjects. Maybe it was a result of the additive character of the problems - only sums of attributes along each path were important. Apparently if the problems did not have this property, the Full Graph displays would have been more useful for the subjects. However in our experiments the subjects did not use any specific option given in the Full Graph display - they neither selected some nodes nor used color coding of individual nodes.

Opinions about the other two types of displays differed from subject to subject. Subject #1 strongly favored the Cumulative Totals display and spent a lot of time exercising the cube magnified option. He also expressed a desire to have a capability to "point" at individual points within the cube. This capability was later implemented. The Record Best display however was not found useful by subject #1.

Subject #2 admitted that he probably had not acquired a full appreciation of the Cumulative Totals display. It seemed to this subject that there were unexplored advantages of using this type of display, which he had probably not understood. However this subject found the Record Best display to be of great use. The subject also expressed a desire to have a capability of restoring a saved solution - its weights as well as its constraints. This capability has been implemented and the subject found it very useful. Both subjects expressed a wish to have all three displays integrated as options in one system.

As the result of these experiments it has been concluded that the enumeration part of the satisficing search is not the most important one for the DM, at least for problems with additive attributes. The 3-D visualization displays are useful but require substantial training before DM utilizes them at full capacity.

It was observed in the experiments that the subjects had realized the relative nature of the weights. The subjects initially expected that the weights were the ultimate factors in finding satisficing decision. After a while they realized that the search required more sophisticated use of the weights together with the constraints.

Two strategies of using the weights/constraints combination were observed. Subject #1 tried to do a coarse selection using the constraints only and then perform a fine selection using the weights. His intention was to narrow down a search region (using the constraints) to a fairly small set of alternatives.

He wanted the reduced search region to be small enough that he could work inside it with the weights only, and at the same time large enough to contain a sufficient number of alternatives. This subject was very upset each time he encountered an infeasible set of constraints. The subject noted that alternatives generated by the computer for an infeasible set of constraints were of no great value to him.

Subject #2 tried to work out a different strategy. He used constraints to limit the values of undesirable attributes and used the weights to increase the values of the desirable ones. However he expressed some doubts as to how to deal with undesirable attributes, namely how to keep them low.

A very important result of the experiments was an observation that the subjects frequently used an explicit utility function of their own. And that utility function was not at all a linear function of the attributes. Or sometimes the subjects preferred to use some functions of the attributes as selection criteria instead of the attributes themselves. For example in Problem #2, "Political Campaign Tour" subject #2 clearly was interested in just two values: TV-time (one of the original attributes) and the sum of Democrats and Independents (a sum of two original attributes). In Problem #6, Time-Sharing Condominium Promotion the same subject was interested in just one value - sum of all coupons divided by total amount of time spent (sum of three original attributes divided by the sum of the other two attributes).

The last example presents additional evidence against using linear functions as valid approximations of one's utility function. It is quite understandable that the problem attributes themselves are not necessarily the criteria the DM would use to select her/his satisficing decision. It seems natural that the DM may wish to use some functions of these attributes as selection criteria. An option to use any functions of the given attributes is planned to be implemented in our system and tested on a human DM.

The sensitivity analysis option has been found useful by both subjects. Both of them had this option turned on all the time. However, subject #1 frequently tried to ignore this option in his weight assignments. He changed all weights within their sensitivity ranges and hoped that another alternative would be selected by the computer because the sensitivity ranges were important if only one weight was changed. This subject also noted that sensitivity of constraints was more useful for him. Subject #2 used the weights sensitivity option very extensively and commented that it was a very useful option.

An unexpected question arose when the subjects were working with the Record Best displays. The GraMAD shows the total values of the paths in bar graph form. A bar for an attribute has zero length when this attribute is at its minimum value. Minimum value was used instead of uniformly zero value in order to increase the resolution of the display (since the DM specifies her/his constraints in the same area of the screen). However as it was asserted by subject #1, that when this bar graph is saved in the Record Best

option it should be saved as a bar graph with bars starting at zero values for all attributes.

This observation agrees well with [Tufte, 1983] where it is noted that having zero length bars corresponding to non-zero values of the attribute is a common misleading way of graphical information presentation. Bar graphs representing different alternatives are used for comparison while total attribute values are shown only to present the computer suggested values of the attributes. Comparison requires the zero origin for all bars so that one could see that, for instance, a value of a particular attribute is twice as big as the value of the same attribute in another alternative decision. This is the reason why in the GraMAD all saved bar graphs have zero values corresponding to zero bar lengths.

All in all the experiments have demonstrated viability of the GraMAD system as a decision aiding tool. Use of *absolute* means, such as constraints, for specifying one's preferences together with *relative* means, such as weights, proved to be a good way to help the DM. Some informal experiments were conducted to test the ideas of the satisficing approach as it was suggested in [Wierzbicki, 1982]. This approach is basically equivalent to eliminating the weights and using the same scheme as is used in the GraMAD treatment of infeasible constraints. It was very simple to restrict the GraMAD to do just that.

It turned out that this method was not the most "convenient" one for the subjects to use. A lack of *relative* means (such as weights) of expressing how to alter an alternative suggested by the computer in the direction most desirable for the DM proved to be a real deficiency of this method. The DM had a hard time trying to select the next reference point in response to the alternatives suggested by the computer.

Graphical arrangement of the GraMAD interface proved to be very useful. The subjects felt that the experiments were fun and enjoyed them, as long as they had interest in the problems they were to solve.

A conclusion could be made from observing subjects developing a "feel" for the problems and performing pairwise comparison of alternatives. Using pairwise comparison as a way of articulating one's preferences is mandated by the classical utility theory. A number of proposed interactive aids for solving MAOP actually use this techniques for soliciting information from the human DM (e.g. [Stewart, 1984]).

However it has been observed that making pairwise comparisons was extremely unpleasant to the subjects, even when the data was presented in a seemingly appealing graphical form (as in the pairwise comparison training). The subjects reported that they actually disregarded the graphical representation and had to perform pairwise comparison of the values for each attribute. This procedure necessitated to do an unpleasant mental number-crunching. After comparing about twenty consecutively presented pairs, subjects' interest usually faded out and they began experiencing fatigues and eye strain.

Our observation of the subjects actually making pairwise comparisons casts substantial doubts on the usefulness of this technique for interactive solution of the MAOP. It was felt that a practically useful human-computer procedure should not involve unpleasant options which are acceptable only for experimental purposes, such as pairwise comparison of alternatives. In fact, the DM would try to avoid using this option if s/he can. However this procedure remains a single *normative* basis for determining one's utility function.

6. REFERENCES

Barrett, D.S., [1980], "A Computer Graphic Method for Visual Search Through Multi-Attribute Data", SM Thesis, M.I.T., Man-Machine Systems Laboratory.

Berkovich, S.Ya., [1977], "The Dimensionality of the Informational Structures in the Space of Perception (Posing of Problem)", Boiphysics, Pergamon Press, pp.1136-1140, September.

Cohon, J.L., [1978], "Multiobjective Programming and Planning", Academic Press, N.Y.

Keeney, R.L. and Raiffa H., [1976], "Decisions with Multiple Objectives: Preferences and Value Tradeoffs", Wiley, New York.

Knepp, L.A., [1981], "Computer Graphics in Multi-Attribute Data Base Searches: Implementation and Evaluation", SM Thesis, M.I.T., Man-Machine Systems Laboratory.

Marcotte, O. and Soland, R.M., [1986], "An interactive branch-and-bound algorithm for multiple criteria optimization", *Managerial Science, Vol. 32, No. 1, pp.61-75.*

Rosenman M.A, and Gero, J.S., [1985], "Reducing the Pareto Optimal Set in Multicriteria Optimization", Engineering Optimization, Vol. 8, pp.189-206.

Rosenthal, R.E., [1985], "Concepts, Theory, and Techniques Principles of Multiobjective Optimization", *Decision Sciences*, vol. 16, pp.133-152.

Sheridan, T.B., [1976], "Toward a General Model of Supervisory Control.", in Sheridan, T.B., Johannsen, G., (Eds.) Monitoring Behavior and Supervisory Control, Plenum Press, N.Y.

Sheridan, T.B. and Ferrell, W.R., [1974], "Man-Machine Systems", M.I.T. Press, Cambridge, Mass.

Stewart T.J., [1984], "Inferring Preferences in Multiple Criteria Decision Analysis Using a Logistic Regression Model", *Management Science*, vol. 30, no. 9, pp. 1067-1077.

Soland, R.M., [1979], "Multicriteria optimization: A general characterization of efficient solutions", *Decision Sciences*, 10, pp. 26-38.

Tufte, E.R. [1983], "The Visual Display of Quantitative Information", Graphics Press, Cheshire, Connecticut

Wierzbicki, A.P., [1982], "A Mathematical Basis for Satisficing Decision Making", *Mathematical Modelling*, Vol.3, pp.391-405.

Yufik, Y.M., [1985], "Intelligent Decision and Training Aid with Optimization Based Inference Engine", in Proceedings of the First Annual AAAI Conference, Dayton, Ohio.

APPENDIX A

Experimental Scenarios Given To the Subjects

INSTRUCTIONS

You are going to participate in a series of experiments designed to study and aid human decision-making. In these experiments you will be acting as a decision-maker. You will need to choose from a large set of possible alternatives. Each alternative will be characterized as a combination of incommensurable attributes, i.e. a tradeoff. Your objective will be to select "the best" (in your opinion) combination as tradeoff.

There will be two types of attributes: positive and negative. It is presumed that you want to maximize values of the positive attributes and to minimize values of the negative ones. However it is impossible to achieve the best performance for all attributes at the same time. So you will have to make some tradeoffs and sacrifice the best available performance for some (or all) attributes. It is up to you which alternative you will finally consider the best one.

In making these choices you will be assisted with several types of computer aids. You will afterwards rate each of these aids as to their usefulness to you. All problems you will be solving have nothing in common and it is desirable that you approach each of them individually without thinking about the problems you have already solved.

You will need to make your decision based solely on the information specified in the description of the problem. You may use your imagination to complete the problem conditions if necessary.

There is no limitation on time you need for solving each problem.

Good luck and thank you very much for your cooperation.

EXAMPLE FOR PRACTICE

This is to introduce you to the substance of the computer aids and to the spirit of the problems that will follow.

You have been living for a couple of years in a third world country. You live high up in the mountains with a research group and your food supply comes every week. Today you will do your shopping at a fruit bazaar. There are several merchants selling different kinds of fruits and vegetables. It is very late and the bazaar is about to close. Each merchant wants to sell you everything that is left. The problem is that you have to run fast from the bazaar entrance to the exit. You must make purchases from each merchant on the path which you select. However you can visit no more than 8 merchants altogether.

You want to buy certain fruits. Each merchant has different amounts of the fruits left. For each merchant you visit, you must buy the specific combination of fruits he offers and pay the price which he asks.

You have a map of the bazaar on the screen. Each merchant is shown as a node of the graph. The bars poking out of each node show total amount of a particular fruit the merchant has available for sale. One of those bars reflects the amount of money the merchant is charging.

You know you must buy the fruits right now. You know that everything you buy will be gratefully consumed by the members of your research group. You also want to have a diversity in your fruit ration, so you don't want to miss any of the available fruits. On the other hand you are trying not to spend too much. So you need to find a best balance of what you can buy at the bazaar.

Assuming that there are 4 positive attributes:

- bananas;
- peaches;
- melons;
- strawberries;

and 1 negative attribute:

- amount of money spent,

try to plan the best (in your opinion) route through the bazaar.

PROBLEM #1: THE MARVELL TRIP

DESCRIPTION

Your company has offered you a vacation trip to the country of Marvell, provided you do a little lobbying while there. Actually you have always wanted to take this vacation and right now have plenty of free time.

You should start and end your route in the same city so that it is going to be a round trip. You have to select all other intermediate cities by yourself. However you may visit no more than 6 cities. All your transportation expenses will be paid for so you do not have to worry about them. A city map of the country is presented to you on the screen.

All cities in the country have similar attractions and are of equal interest to you. However, the bonuses which you get in each city vary from city to city.

In each city you get the following things for FREE:

- several nights at ANY hotel of your choice;
- several meals at ANY restaurant of your choice;
- several paid admissions to ANY theater/concert show of your choice;

However, in each city in order to exercise your free options, you will have to spend some time standing outdoors in long lines. This time

varies from city to city. Also as a part of the deal for each city you are required to attend several boring gatherings in crowded halls on behalf of local politicians. This time of course depends on the city.

Otherwise you can have as much fun as you wish. To help you to assess relative values of all bonuses, we may mention that all cities in Marvell are very similar to Boston -- in size, lifestyle and prices.

PROCEDURE

There are 5 attributes altogether for each city:

Positive:

nights at hotels;
meals at restaurants;
paid admittances to theater/concert shows

Negative:

time spent outdoor waiting in lines;
time spent indoor at gatherings with local politicians.

PROBLEM #2: POLITICAL CAMPAIGN TOUR

DESCRIPTION

You decided to run for a major political office. At present you are planning a campaign trip trough a very important region. You know where your route starts as well as where it will end. It is also possible to visit only up to 7 cities. Your itinerary should be planned with an objective of selecting intermediate stops so that you will benefit as much as possible from this tour. You have your own political platform (whatever it is) and you hope to attract voters registered with any political party, as well as independents. It is up to you to decide how likely these voters are to accept your position. In order to win the elections you need a simple majority of votes. Your appearance in each city will be covered by national television, but the amount of time you receive depends varies from city to city.

You know the following information about each city:

- number of registered Republicans living in the area;
- number of registered Democrats living in the area;
- number of registered Independents living in the area;
- number of unregistered voters living in the area;
- amount of national TV time you receive.

To help you to assess the data we note that the population of the potential voters watching national TV is about 50 times greater than the average population of the cities in the region you visit.

PROCEDURE

There are 5 attributes for each city.
All of them are positive:
number of registered Republicans living in the area;
number of registered Democrats living in the area;
number of registered Independents living in the area;
number of unregistered voters living in the area;
amount of national TV time you receive.

PROBLEM #3: BOSTON SURPRISES

DESCRIPTION

The City of Boston has allocated some money for a special award bonus. They will give you a Boston city map. Then you need to select a route from a node marked START to the node marked END. You can visit up to 5 nodes on your way. In each stop of your path you will get a bag full of goodies. By taking all prizes inside a bag you also agree to fulfill all obligations required for taking the bag.

Each bag contains certain amounts of the following "goodies":

- several nickels (in number of real coins);
- coupons for ice cream (in oz);
- coupon for any drink of your choice (in maximal amount of money);

When you select a bag you will also be required to do the following:

- put a thread through a needle several times (in number of times);
- stand up straight, without walking, talking, reading, or watching anything (in minutes);

This game will be for real. You will actually be handed out all pleasant attributes from a bag which you select as the best one. You also will have to perform specified amounts of all negative attributes for this bag.

PROCEDURE

There are 5 attributes altogether in each bag:

Positive:

nickels;
ice cream coupons;
drink coupons;

Negative:

putting a thread through a needle;
standing up straight and doing nothing;

PROBLEM #4: MIT PRIZE RUN

DESCRIPTION

MIT has awarded you a special bonus. They will give you an MIT map. Then you need to select a route from a node marked START to the node marked END. You can visit up to 6 nodes on your way. In each node of your path you will collect certain prizes, but also will accumulate some obligations. By selecting a route you agree to fulfill all obligations accumulated along this route, in order to get all prizes.

The following prizes will be handed out in each node:

- candies of your choice (in number of pieces, each worth no more than 15c);
- any (available for sale locally) fruits of your choice (in oz);
- Swiss chocolate (in grams);

You will have to do the following things in return:

- to do several knee-bends;
- to memorize certain number of lines of poetry.

This scenario will actually be played out. After your final decision is made, you will really be handed out all collected attributes along the path you select as the best. On the other hand you will have to perform accumulated amounts of obligations for that path.

PROCEDURE

There are 5 attributes altogether for each path:

Positive:

candies;
fruits;
Swiss chocolate;

Negative:

knee-bends;
memorizing poetry.

PROBLEM #5: CAMBRIDGE WANDER PACKAGES

DESCRIPTION

The City of Cambridge has offered you a large choice of "wander packages". You will get a Cambridge city map. Then you need to select a route from a node marked START to the node marked END. You can visit up to 6 nodes on your way. In each node of your path you will collect a package. By accepting the goodies inside a package you also agree to fulfill all obligations required while taking the package.

Each package contains certain amounts of the following "goodies":

- coupons for stationary from the COOP (in maximum amount of money);
- any pastry of your choice (in grams);
- any cold cuts of your choice (locally available) (in oz);

By selecting a package you will also be required to do the following things:

- read out loud a given poem, 12 lines long (number of times);
- jump up down on one foot (number of times);

Remember that this game is for real. You will actually be handed out all things from a package you select.

PROCEDURE

There are 5 attributes altogether for each package:

Positive:

coupons for stationary from the COOP;
pastry of your choice;
cold cuts of your choice;

Negative:

reading out loud a given poem;
jumping up/down on one foot;

PROBLEM #6: TIME-SHARING CONDOMINIUM PROMOTION

DESCRIPTION

You have received a bunch of letters from very aggressive time-sharing condominium promoters. After reading the Consumers Report magazine you have decided that these things were a hoax. However it is a summer, you have plenty of free time and not too many things to do. So you have decided to take advantage of the promotions and have as much fun as possible.

The sites of the condominiums are located in a big area of New Hampshire and you have decided to visit at most 5 different sites. You have prepared a map (it is displayed on the screen) of the area and now you are planning the route.

At each condominium site you will have to subject yourself to a tour of the facility accompanied with a boring sales pitch prepared for very naive people (unlike yourself).

Because of a great demand on the promotion, there is a great deal of traffic going around each site. Practically all driving time will be spent

entering and leaving the sites. It is a fairly hot summer. so decide for yourself how pleasant it is to drive in that heat! All other driving - from site to site takes practically no time in comparison with entering and leaving the sites.

And now at last the pleasant part of this undertaking (since your are not going to buy the condos). At each site you will be rewarded with special coupons (or credits) which have monetary denomination. These coupons are redeemable for goods at a special store (which does not accept normal money). This store is very well stocked and the inventory and prices are similar to those at the Harvard COOP.

You are going to get coupons for goods at 3 different departments:
- books and records;
- clothes and shoes;
- appliances, electronics and computers.

These coupons are redeemable only in the department they are issued . . 'ou can get anything in the department for a sum not exceeding the total va of your coupons. You consider that the coupons will be worth something to you, so it may well be worth spending some time engaging in those boring activities, the condo site tours and driving in and out of the sites.

PROCEDURE

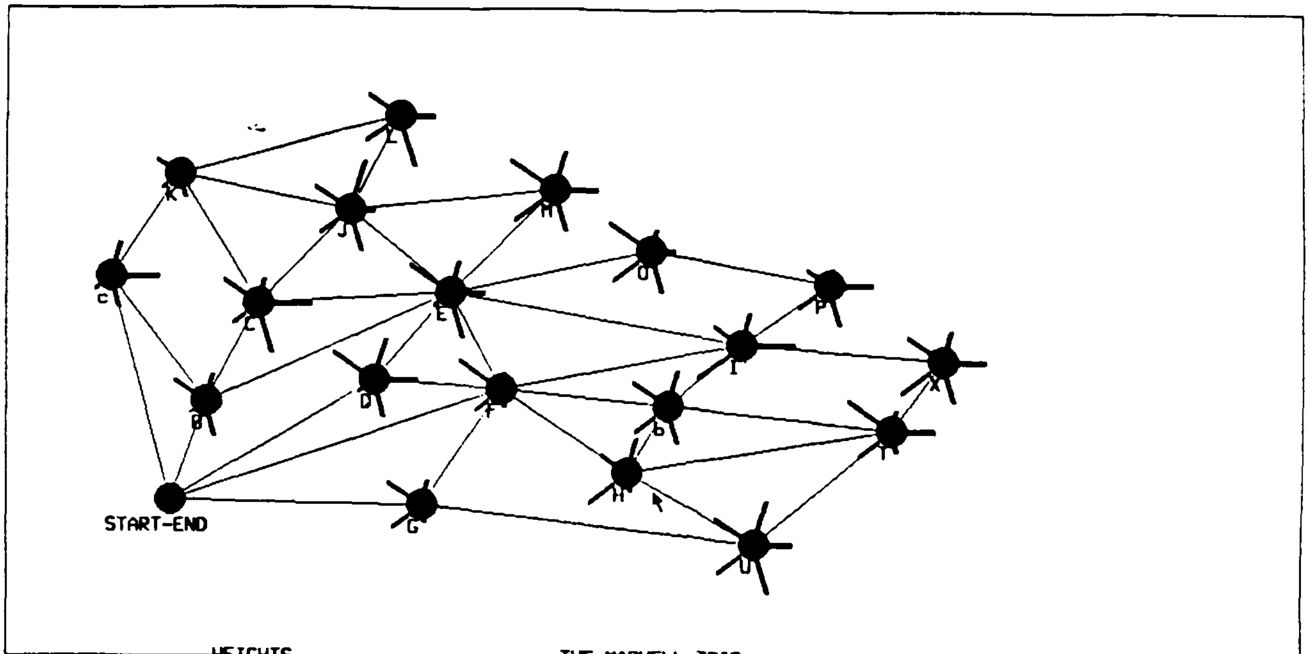
There are 5 attributes altogether for each site:

Positive:

amount of free coupons for books and records;
amount of free coupons for clothes and shoes;
amount of free coupons for appliances, electronics and computers.

Negative:

time spent touring the site and listening to sales pitches;
time spent driving in and out of the sites.



WEIGHTS — THE MARVELL TRIP — MIN — ACCUMULATED TOTALS — MAX

		MIN	MAX
20.0%	Gatherings: 0-42 (hours)	8	182
20.0%	Waiting: 1-29 (hours)	4	126
20.0%	Theaters: 0-19 (admissions)	12	97
20.0%	Restaurants: 0-19 (meals)	8	69
20.0%	Hotels: 0-8 (nights)	0	27

0 10 20 30 40 50 60 70 80 90 100

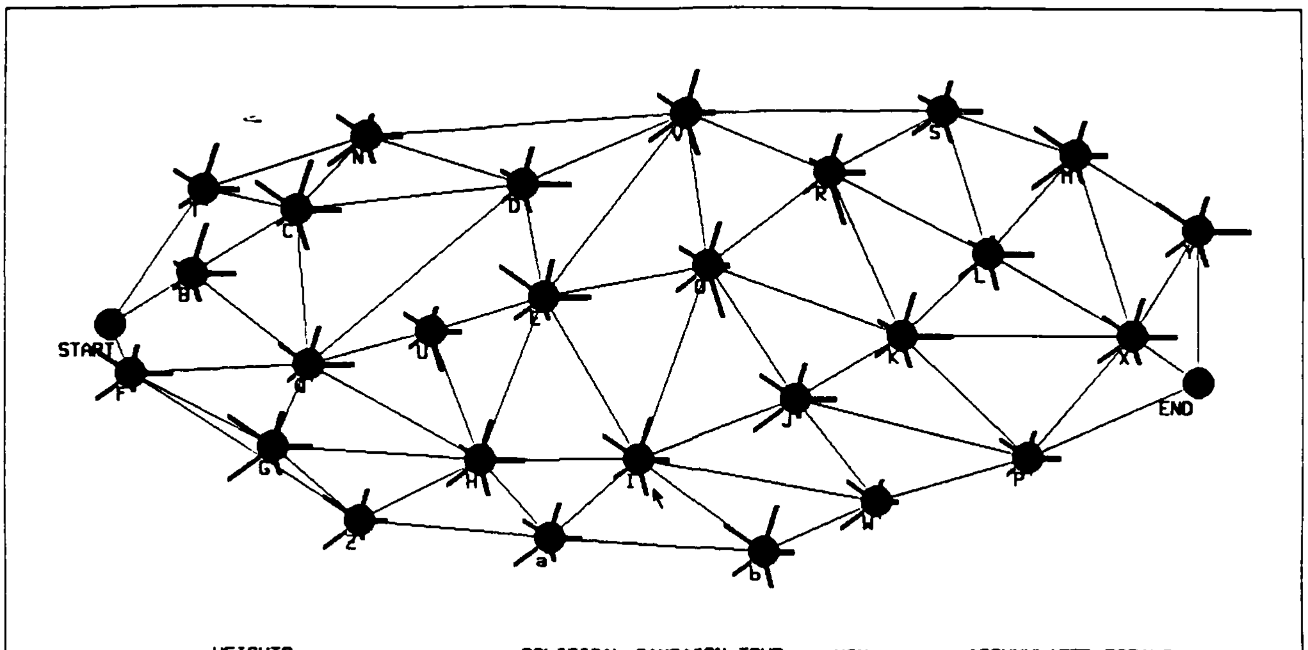
BEST PATH Visit up to 6 nodes

CONSTRAINTS ARE INFEASIBLE

SENSITIVITY OFF

Total Pareto paths: 84

Total paths: 101



WEIGHTS — POLITICAL CAMPAIGN TOUR — MIN — ACCUMULATED TOTALS — MAX

WEIGHTS		MIN	MAX
20.0%	TV time: 0-9 (minutes)	12	36
20.0%	Unregistered: 24-113 (thousand)	313	544
20.0%	Independents: 19-117 (thousand)	305	562
20.0%	Democrats: 16-97 (thousand)	375	569
20.0%	Republicans: 10-58 (thousand)	205	300

0 10 20 30 40 50 60 70 80 90 100

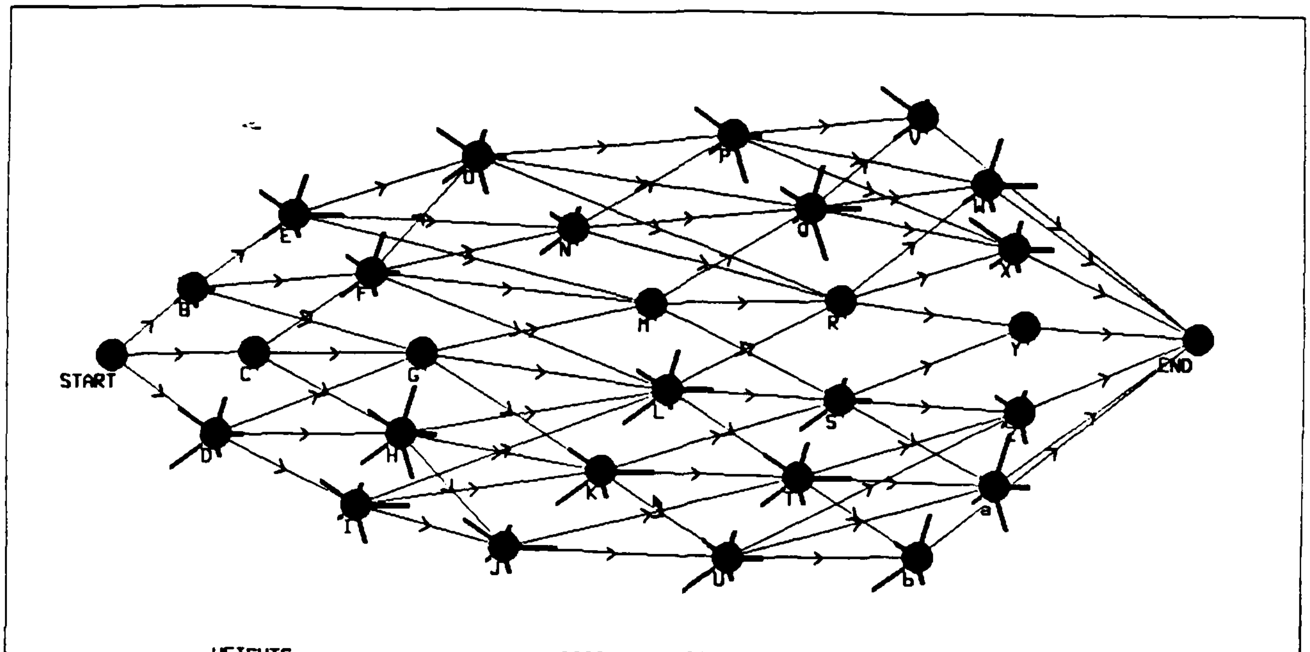
CONSTRAINTS ARE INFEASIBLE

BEST PATH Visit up to 7 nodes

SENSITIVITY OFF

Total Pareto paths: 68

Total paths: 255



WEIGHTS	BOSTON SURPRISES	MIN	ACCUMULATED TOTALS	MAX
20.0%	Stand up: 0-10 (minutes)	0		24
20.0%	Thread: 0-14 (times)	0		65
20.0%	Drink: 0-79 (cents)	0		270
20.0%	Ice cream: 0-19 (fl.oz)	0		74
20.0%	Nickels: 0-11 (number)	0		43

0 10 20 30 40 50 60 70 80 90 100

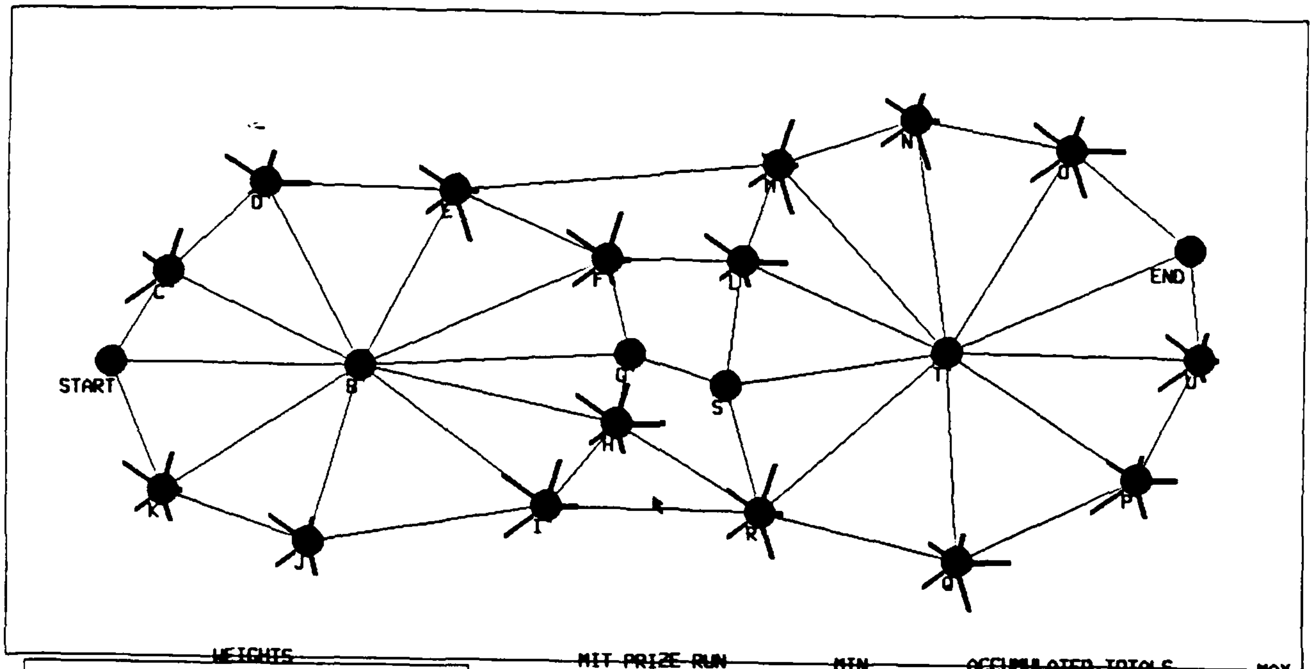
BEST PATH Visit up to 5 nodes

CONSTRAINTS ARE INFEASIBLE

SENSITIVITY OFF

Total Pareto paths: 144

Total paths: 252



WEIGHTS — MIT PRIZE RUN — MIN — ACCUMULATED TOTALS — MAX

		MIN	MAX
20.0%	Memorizing: 0-10 (lines)	0	69
20.0%	Knee-bends: 0-26 (number)	0	109
20.0%	Chocolate: 0-94 (grams)	0	399
20.0%	Fruits: 0-28 (oz)	0	102
20.0%	Candies: 0-9 (pieces)	0	28

0 10 20 30 40 50 60 70 80 90 100

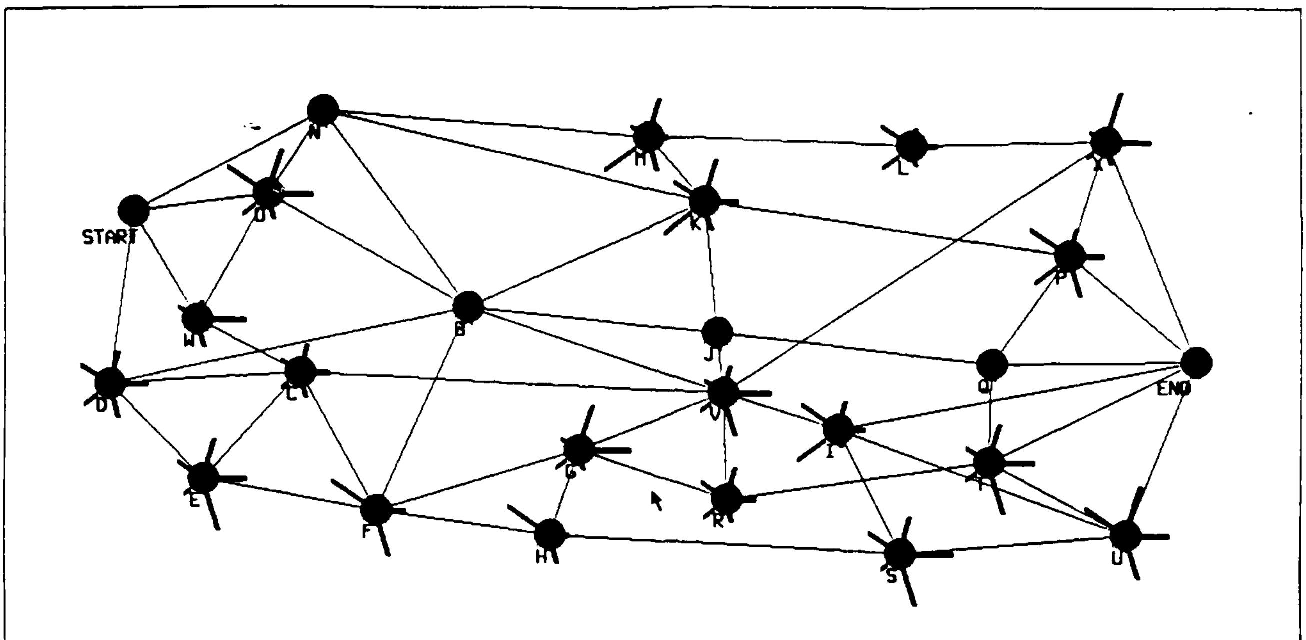
BEST PATH Visit up to 6 nodes

CONSTRAINTS ARE INFEASIBLE

SENSITIVITY OFF

Total Pareto paths: 86

Total paths: 210



WEIGHTS	CAMBRIDGE WANDER PACKAGES	MIN	ACCUMULATED TOTALS	MAX
20.0%	Jump: 0-14 (times)	0		63
20.0%	Read loud: 0-10 (times)	0		32
20.0%	Cold cut: 0-19 (oz)	0		84
20.0%	Pastry: 0-79 (grams)	0		299
20.0%	Stationary: 0-107 (cents)	0		472

0 10 20 30 40 50 60 70 80 90 100

BEST PATH Visit up to 6 nodes

CONSTRAINTS ARE INFEASIBLE

SENSITIVITY OFF

Total Pareto paths: 102

Total paths: 324

APPENDIX B

Plots of Experimental Data

This Appendix has been removed from the manuscript due to its length. All plots are available at the Man-Machine Systems Laboratory.

APPENDIX C

Instructions for a User of the Satisficing Computer Aid GraMAD[1]

I. FULL MAP DISPLAY

C.I.1 INTRODUCTION

The objective of the aid is to help you find a satisfactory path connecting the marked starting node with the marked ending node. Fig.C-1 shows the initial computer-generated display. The path must go through the designated possible connections between the nodes. Note that some of the connections are bi-directional (two-way) while some are one-way only. Your path should not go through more than the limiting number of nodes (as it is specified in the middle of the lower half of the screen). Any path which satisfies these conditions is called *feasible*.

In order to compare alternative paths, you ought to be concerned with values of several parameters, or *attributes*, associated with each path. These attributes are listed in the center of the lower half of the screen along with units of measure of each attribute. (All information related to an attribute is consistently presented on the screen using a color assigned to this attribute.) A stop at any node independently "scores" values of each attribute. A score of an attribute along any path is a sum of scores of this attribute in all nodes the path goes through. From the nature of the attributes it is obvious that some of them are *desirable* (the more - the better) while others are *undesirable* (the less - the better).

Color bars poking out of a node reflect amounts of attributes (corresponding to the colors) scored at this node. Note that all scored values are positive. The longer the bar - the higher the value of a corresponding attribute (regardless whether the attribute is desirable or undesirable).

There is a finite number of feasible paths. (The exact number of feasible paths is displayed in the bottom of the screen.) Among all *feasible* paths there is possibly one which delivers the ***maximal*** value of the attribute 1 (the cost). There is another path which has the ***minimal*** value of the cost. A value of the cost of any other path would therefore lie between these

1 These instructions are illustrated with some photos taken from the display screen. Unfortunately the quality of colors on the pictures is much inferior to the quality of the colors on the actual computer display.

minimal and maximal values. The same holds for all other attributes. These minimal and maximal values of each attribute are displayed in the right lower part of the screen.

C.I.2 CONTROL ACTIONS

You can independently change the following parameters: a) weights; b) constraints; c) nodes necessary to visit. At any time you can request the computer to show you the best path which takes previously specified weights, constraints and desirable nodes into account. You can also employ the sensitivity analysis option when asking the computer to show the best path.

All control actions are performed by moving the cursor (by means of moving the mouse) into designated areas of the screen and pressing any button on the mouse.

a) WEIGHTS

The weight area is located in the lower left part of the screen. The weights are changed by moving the cursor into the weight-bar area of the screen and by pressing a mouse button. A current value of the corresponding weight is displayed to the right of the bar.

The bars can be changed independently and each weight can be given any value between 0 and 99.9%. Since the sum of all weights must always be 100%, the computer has to re-normalize the weights after they have been changed. Every time the weights are changed, a flashing message "WEIGHTS ARE CHANGED" appears on the screen and no normalization is performed during that time (see Fig. C-2). When you decide that the weights have been adequately altered, the cursor must be moved outside the weight area of the screen and any mouse button must be depressed at least once. At that instant the computer will re-normalize the weights so that they sum to 100% (see Fig. C-3).

When the computer re-normalizes the weights, the following algorithm is used. If only a few (not all) weights have been changed, and the sum of new values of the changed weights is less than 100%, then these weights stay intact. Then non-changed weights are re-scaled so that the sum of all weights is 100%, while the relative values of the non-changed weights (as compared to each other) remain the same. If the sum of the new values of the weights is greater than 100% or all the weights have been altered, the computer re-scales all the weights while keeping their relative values the same. After re-normalizing the weights the computer turns the flashing message off.

The brightness of the nodes on the map is adjusted so that the brighter the node is - the more desirable it is in respect to the current values of weights.

In order to reset all weights to equal values one should press I (stands for *initialize*) key.

b) CONSTRAINTS

The constraint area is located in the lower right part of the screen. Each constraint is independently adjusted by moving the cursor in the constraint-rectangle area and then pressing any mouse button. Note that if an attribute is *desirable* (the more its value - the better) only the lower constraint is available for adjustment. Alternatively if an attribute is undesirable (the less its value - the better) only the upper constraint could be adjusted. (It is presumed that "good" things should not be limited from above, while "bad" things could score as little as possible and should not be restricted from below.).

A number to the left of any constraint-rectangle shows the current lower value of the lower constraint on this attribute. A number to the right of the bar shows the upper constraint on the attribute. (See Fig. C-4)

c) NODES NECESSARY TO VISIT

Any node can be selected for a visit. In order to select/unselect a node, the cursor must be moved inside the node circle and any mouse button depressed. If the node was not previously selected, a colored rectangle will appear around the node and it will become selected. If there is no path which could go through previously selected nodes and the node which is attempted to be selected, a message "NODE CANNOT BE SELECTED" appears on the screen and the selection fails. If the node has been previously selected (already was surrounded with a colored rectangle), it will become unselected. (See Fig. C-5)

C.I.3 BEST PATH

After the weights and the constraints have been adequately specified you may ask the computer to show the "best path". This is done by moving the cursor inside the "BEST PATH" rectangle and then pressing any mouse button. The computer will 1) consider only those paths which go through selected nodes, and 2) try to satisfy the constraints on total values of the attributes. If there are paths which satisfy these constraints, the computer will display the best one (in a sense of a weighted sum of total attribute values). (See Fig. C-6)

If there are no paths which satisfy the specified constraints, the computer will find a path with the total attribute values which are as close as possible to the specified constraint values. (The closeness will be measured using the specified weights). A flashing message "CONSTRAINTS ARE INFEASIBLE" will appear on the screen (below the constraints area). (See Figs. C-7,C-8)

The number appearing in the "BEST PATH" rectangle indicates a total number of paths which have been considered by the computer. Total cumulative values of the attributes corresponding to this path are displayed in the constraint area (in the form of a bar-graph). (You can see that the difference between Fig. C-7 and Fig.C-8 is that there were no selected nodes on the display Fig. C-7, and the computer had to search 196 different paths. On the display Fig. C-8 there were three nodes selected and the number of paths had been reduced to 38.)

In order to get rid of the currently displayed best path, the cursor must be moved inside the "Best Path" rectangle and any mouse button must be depressed.

C.I.4 SENSITIVITY ANALYSIS

In order to turn the sensitivity analysis option on, the cursor must be moved inside the "SENSITIVITY OFF" rectangle and any mouse button depressed. Similarly for turning this option off: the cursor must be moved inside the "SENSITIVITY ON" rectangle and a mouse button depressed.

The sensitivity analysis allows one to view what minimal changes of the weights must be in order to get a different best path. Whenever a best path is displayed, the weight bars (in the weights area) and the constraint bars (in the constraints area) become surrounded by wider rectangles of the same color. A different best path (advised by the computer) may be obtained by changing only one weight beyond the area confined inside a corresponding wide rectangle (See Figs. C-7,C-8).

The wide sensitivity rectangles around the constraint bars appear only when the constraints are infeasible. These sensitivity rectangles show how a *feasible* constraints set is obtained by changing only one constraint.

II. COMPLETE SET OF CUMULATIVE TOTALS DISPLAY

C.II.1 INTRODUCTION

Fig.C-9 shows the initial computer-generated display, which has the same control of weights, constraints and sensitivity on/off option as the "Full Map" presentation. The best path is displayed as a flashing (red-to-blue) dot inside the cubes.

Every alternative path can be considered as a point in a multi-dimensional space of the attributes. However more than 3 dimensions cannot be visualized by humans. On the display are small windows with small cubes representing all possible combinations of three attributes out of the set of all attributes. These cubes can be rotated about any axis, and magnified.

C.II.2 ROTATING AND ORIENTING THE CUBES

In order to rotate any cube, the cursor must be kept within a small window area of the cube and a mouse button must be depressed. Three mouse buttons represent three axes of rotation.

Several keys on the computer keyboard are used for controlling the rotation of the cubes:
Depressing the "+" key increases the rotation speed. The "-" key decreases the rotation speed.
The "o" key pressed at the same time as a mouse button causes the cube to oscillate (enhancing the 3-D effect).
The "r" key reverses the direction of rotation.
The "1", "2' and "3" keys re-orient the 3-D cube so that it looks like a 2-D square (getting rid of one dimension of the cube).

Fig. C-10 shows one cube being rotated. Fig. C-11 shows a best path display after several cubes have been rotated. Note that both the weights and the constraints have been changed.

C.II.3 MAGNIFYING THE CUBES

In order to magnify a small cube, the cursor must be moved into the small cube window and the "b" (stands for "big") key on the keyboard must be hit once. All the above mentioned rotation controls are the same for the magnified cube. In order to return to the original multi-cube display, the "b" key must be hit once again. (See Fig. C-12)

Fig. C-13 shows the magnified cube being re-oriented with only one plane visible after hitting "1" key on the keyboard (the dimension of the attribute #1 - Chemical X - has disappeared). Fig. C-14 shows the result of hitting the key "2" (the dimension of the attribute #2 - Gas Y - has disappeared).

C.II.4 SELECTING A POINT INSIDE THE CUBE

Any point within the magnified cube can be selected and its attribute values will be displayed. In order to select a point, the cursor must be moved.

C.II.5 VISUALIZING THE CONSTRAINTS

This option works only with a magnified cube. The constraints can be visualized as a colored parallelepiped (confining the feasibility region) inside the big cube (See Fig. C-15). In order to turn this option on, the cursor must be moved inside the "SHOW CONSTRAINTS" rectangle and any mouse button must be pressed. (The same procedure turns this option off.)

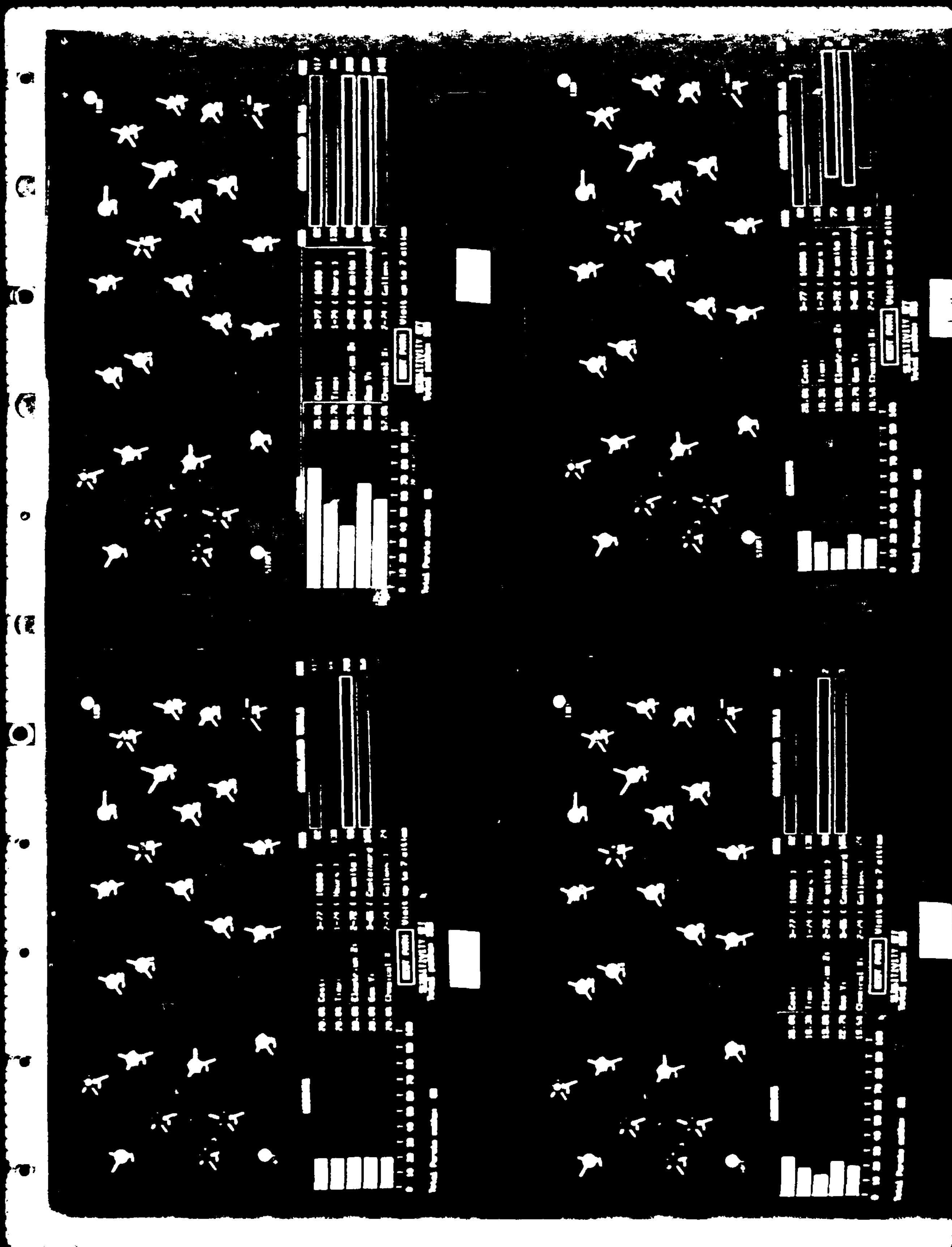
Fig. C-16 shows a big cube with constraints visualized, displayed in motion.

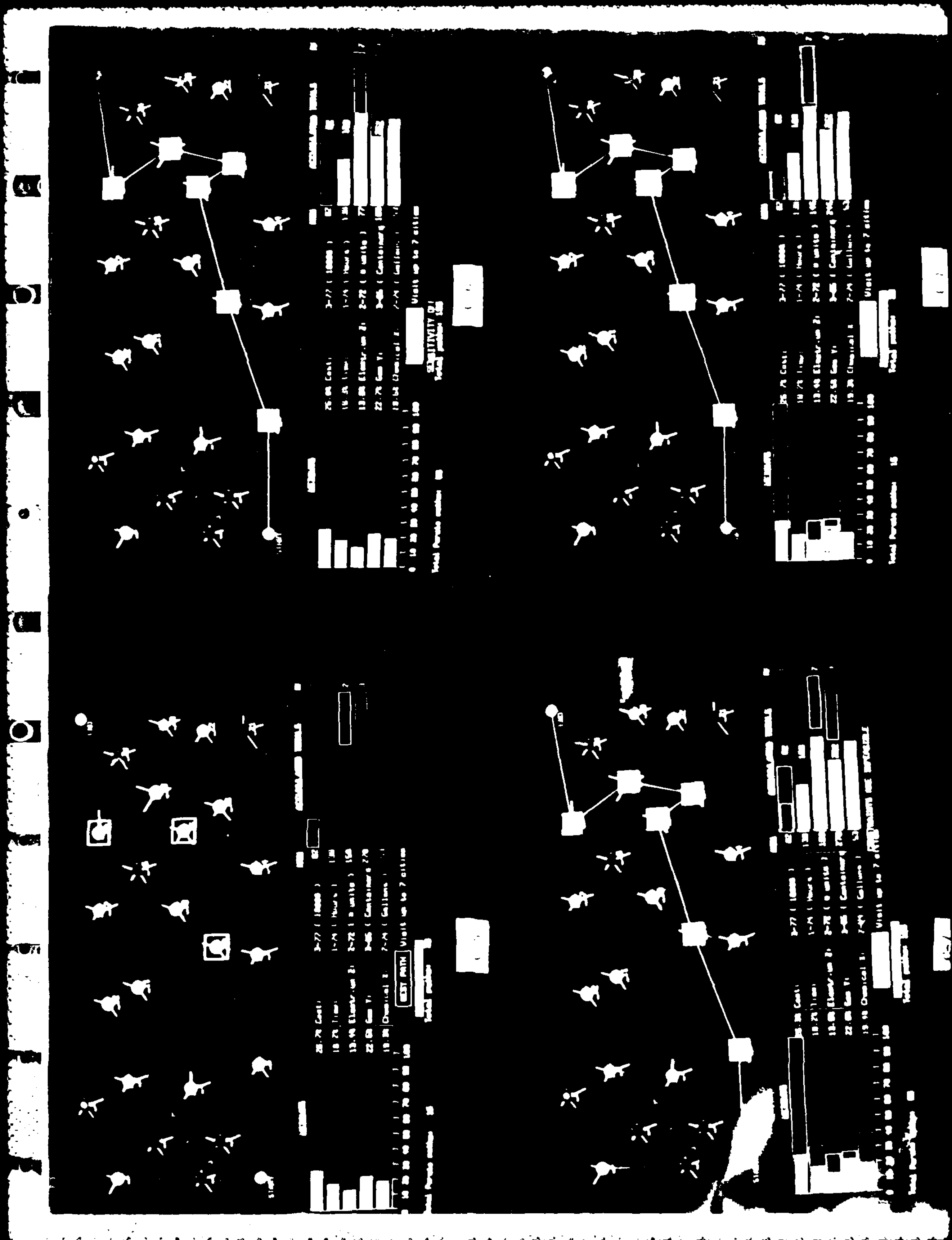
III. RECORD OF BEST CUMULATIVE TOTALS DISPLAY

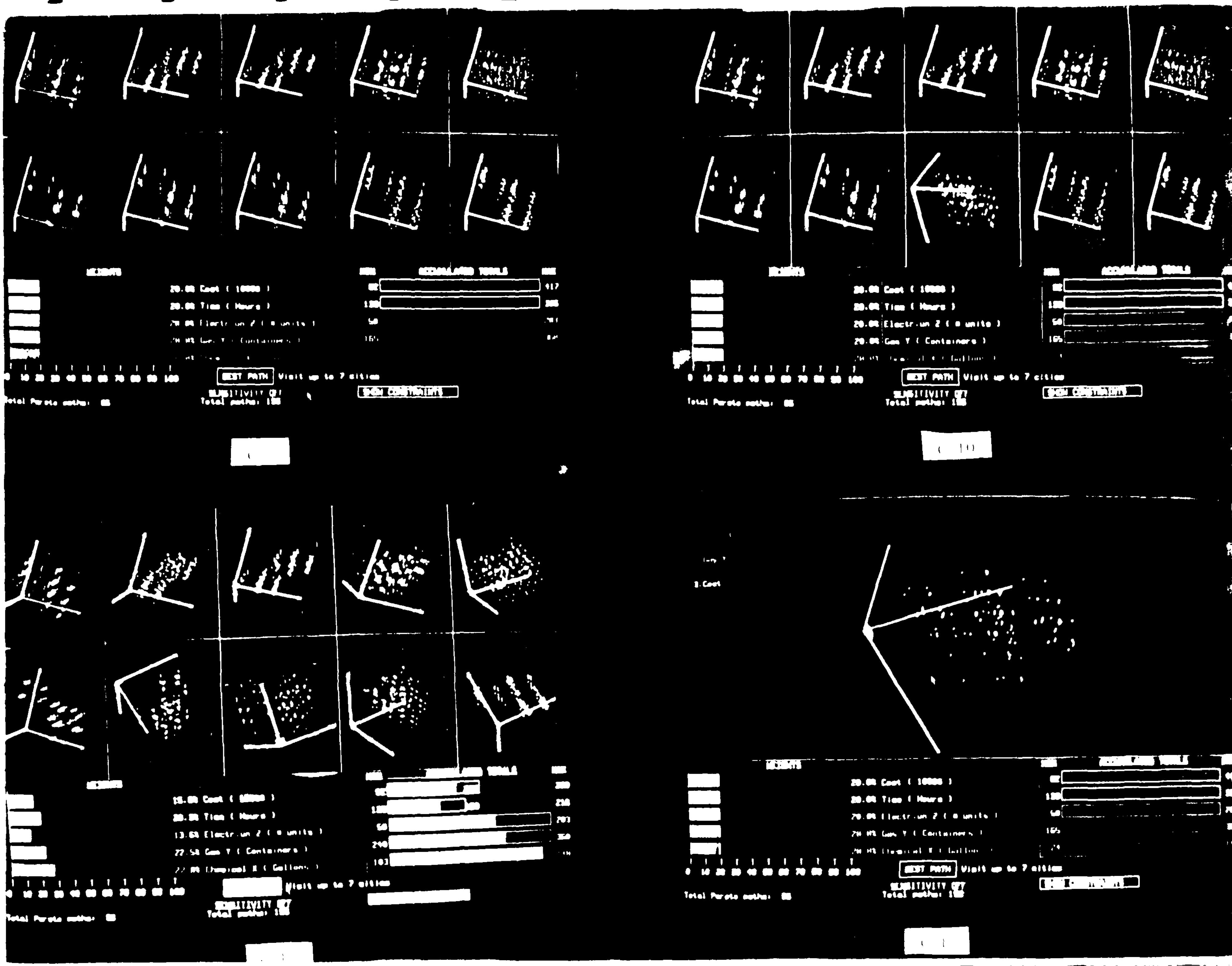
This display mode employs the same control of weights, constraints and sensitivity on/off option as does the "Full Map" presentation. The lower portion of the screen is also the same.

Any time the computer presents its best path suggestion, an image of the cumulative totals bar-graph for this path can be saved in the upper part of the screen. This is done by moving the cursor inside the "SAVE" rectangle and pressing any mouse button.

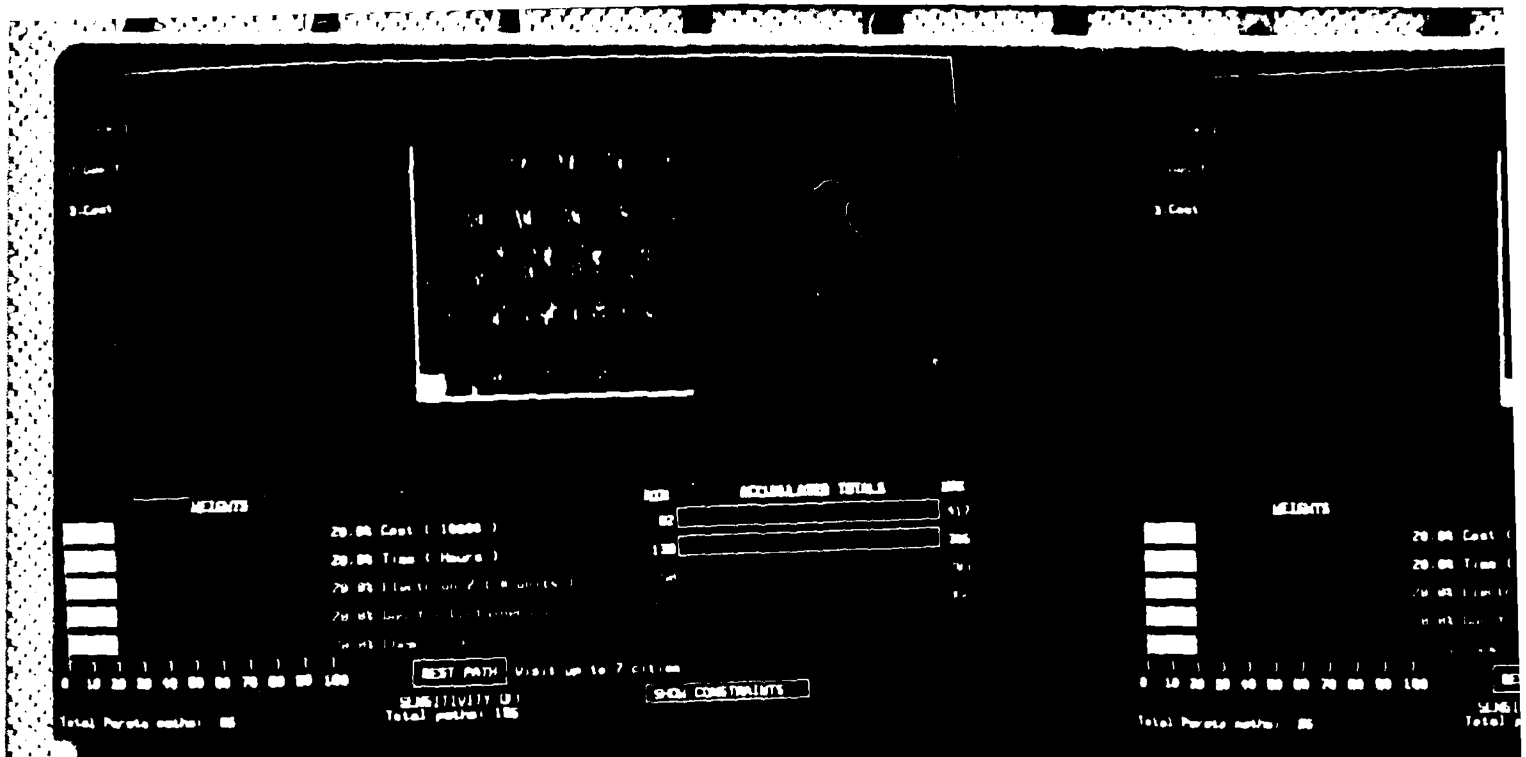
The screen can hold up to nine (though this number can be easily changed) saved bar-graphs (See Fig. C-17). If the attempt to save a new bar-graph is made after the screen already has been filled out, the new bar-graph will replace one of the previously saved ones. The cursor changes its shape and the computer waits to be shown the old bar-graph to be replaced. The cursor must be moved inside a bar-graph to be replaced and any mouse button must pressed. The new bar-graph (cumulative totals of the currently displayed best path) then replaces the old bar-graph.







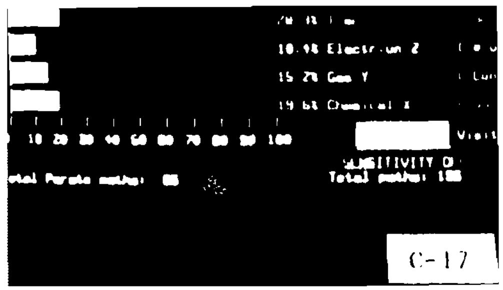
WEIGHTS
ACCUMULATED TOTALS
20.00 Cost (10000)
20.00 Time (Hours)
Electr.un Z (# units)
Gas Y (Containers)
Chemical X (Gallons)
0 10 20 30 40 50 60 70 80 90 100
BEST PATH
Visit up to 7 cities
SHOW CONSTRAINTS
SENSITIVITY OFF
Total Pareto paths:
Total paths:


WEIGHTS
20.0% Cost (10000)
20.0% Time (Hours)
ACCUMULATED TOTALS
BEST PATH
Visit up to 7 cities
SHOW CONSTRAINTS

C-13


ACCUMULATED TOTALS
20.0% Cost (10000)
20.0% Time (Hours)
20.0% Electron Z (# units)
20.0% Gas Y (Containers)
20.0% Chemical X (Gallons)
Visit up to 7 cities

C-15



0 10 20 30 40 50 60 70 80 90 100
SENSITIVITY OF

DISTRIBUTION LIST

Engineering Psychology Program
Office of Naval Research
Code 1142EP
800 North Quincy Street
Arlington, VA 22217-5000

J. Randy Simpson
Statistics Program Code 1111SP
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000

Dr. Lyle D. Broemeling
Code 1111SP
Office of Naval Research
800 N. Quincy Street
Arlington, VA 22217-5000

Information Sciences Division
Code 1133
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000

Dr. Randal P. Schumaker
NRL A. I. Center
Code 7510 R & D Command
Naval Research Laboratory
Washington, D.C. 20375-5000

Director
Technical Information Division
Code 2627
Naval Research Laboratory
Washington, D.C. 20375-5000

Dr. Michael Melich
Communications Sciences Division
Code 7500
Naval Research Laboratory
Washington, D.C. 23075-5000

Dr. A. F. Norcio
Computer Sciences & Systems
Code 7592
Naval Research Laboratory
Washington, DC 20375-5000

Human Factors Department
Code N-71
Naval Training Systems Center
Orlando, FL 32813

Mr. Norm Beck
Combat Control Systems Department
Code 35
Naval Underwater Systems Center
Newport, RI 02840

Human Factors Engineering
Code 441
Naval Ocean Systems Center
San Diego, CA 92152

Dr. Gary Poock
Operations Research Department
Naval Postgraduate School
Monterey, CA 93940

Dr. William Uttal
Naval Ocean Systems Center
Hawaii Laboratory
P. O. Box 997
Kailua, HI 96734

Dr. L. Chmura
Computer Sciences & Systems
Code 7592
Naval Research Laboratory
Washington, D.C. 20375-5000

CDR C. Hutchins
Code 55
Naval Postgraduate School
Monterey, CA 93940

Dr. Stanley Collyer
Office of Naval Technology
Code 222
800 North Quincy Street
Arlington, VA 22217-5000

Professor Michael Sovereign
Joint Command, Control and
Communications Curriculum
Code 74
Naval Postgraduate School
Monterey, CA 93943

Aircrew Systems Branch
Systems Engineering Test
Directorate
U.S. Naval Test Center
Patuxent River, MD 20670

CAPT Robert Biersner
Naval Biodynamics Laboratory
Michoud Station
Box 29407
New Orleans, LA 70189

Dr. George Moeller
Human Factors Engineering Branch
Naval Submarine Base
Submarine Medical Research Lab.
Groton, CT 06340

Commanding Officer
Naval Health Research Center
San Diego, CA 92152

Dr. Jerry Tobias
Auditory Research Branch
Submarine Medical Research Lab
Naval Submarine Base
Groton, CT 06340

LCDR T. Singer
Human Factors Engineering Division
Naval Air Development Center
Warminster, PA 18974

LCDR R. Carter
Office of Chief on Naval Operations
(OP-01B)
Washington, D.C. 20350

CDR W. Moroney
Naval Air Development Center
Code 602
Warminster, PA 18974

Human Factors Branch
Code 3152
Naval Weapons Center
China Lake, CA 93555

Dr. Eugene E. Gloye
ONR Detachment
1030 East Green Street
Pasadena, CA 91106-2485

Dr. Robert A. Fleming
Human Factors Support Group
Naval Personnel Research
and Development Center
1411 South Fern Street
Arlington, VA 22202

Dr. Edgar M. Johnson
Tecnical Director
U.S. Army Research Institute
Alexandria, VA 22333-5600

Director, Organizations and Systems
Research Laboratory
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

Dr. Milton S. Katz
Director, Basic Research
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

Dr. A. Fregly
U.S. Air Force Office of
Scientific Research
Life Science Directorate, NL
Bolling Air Force Base
Washington, D.C. 20332-6448

DISTRIBUTION LIST

Dr. Earl Alluisi
Chief Scientist
AFHRL/CCN
Brooks Air Force Base, TX 78235

Dr. J. Tangney
Directorate Life Sciences
AFSOR
Bolling AFB
Washington, D.C. 20032-6448

Mr. Yale Smith
Rome Air Development
Center, RADC/COAD
Griffiss AFB
New York, NY 13441-5700

Dr. Alan Leshner
Deputy Division Director
Division of Behavioral and
Neural Sciences
National Science Foundation
1800 G. Street, N.W.
Washington, D.C. 20550

Defense Technical Information
Center (2)
Cameron Station, Bldg. 5
Alexandria, VA 22314

Dr. Amos Tversky
Dept. of Psychology
Stanford University
Stanford, CA 94305

Dr. Donald D. Hoffman
University of California
(Irvine)
School of Social Sciences
Irvine, CA 92717

Dr. Stanley Deutsch
NAS-National Research Council
(COFH)
2101 Constitution Avenue, N.W.
Washington, D.C. 20418

Dr. Deborah Boehm-Davis
Department of Psychology
George Mason University
4400 University Drive
Fairfax, VA 22030

Dr. James T. Todd
Brandeis University
Waltham, MA 02254

Dr. David Van Essen
California Institute of Technology
Division of Biology
Pasadena, CA 91125

Dr. James H. Howard, Jr.
Department of Psychology
Catholic University
Washington, D.C. 20064

Dr. William Howell
Department of Psychology
Rice University
Houston, TX 77001

Dr. Christopher Wickens
Department of Psychology
University of Illinois
Urbana, IL 61801

Dr. Robert Wherry
Analytics, Inc.
2500 Maryland Road
Willow Grove, PA 19090

Dr. Lola L. Lopes
Department of Psychology
University of Wisconsin
Madison, WI 53706

Dr. Stanley N. Roscoe
New Mexico State University
Box 5095
Las Cruces, NM 88003

Dr. Marvin Cohen
Decision Science Consortium, Inc.
Suite 721
7700 Leesburg Pike
Falls Church, VA 22043

Dr. Scott Robertson
Catholic University
Department of Psychology
Washington, DC 20064

Dr. Andrew P. Sage
Assoc. V.P. for Academic Affairs
George Mason University
4400 University Drive
Farifax, VA 22030

Dr. James Ballas
Georgetown University
Department of Psychology
Washington, DC 20057

Dr. John Payne
Graduate School of Business
Administration
Duke University
Durham, NC 27706

Dr. Tomaso Poggio NE 43-787
MIT
Center for Biological Information
Processing
Cambridge, MA 02139

Dr. Whitman Richards
MIT E10-120
Dept. of Psychology
Cambridge, MA 02139

Dr. Robert A. Hummel
New York University
Courant Inst. of Mathematical
Sciences
251 Mercer Street
New York, NY 10012

Dr. Kent A. Stevens
University of Oregon
Dept. of Computer & Info. Sci.
Eugene, OR 97403

Dr. Donald A. Glaser
U. of California, Berkeley
Department of Molecurar Biology
Berkeley, CA 94720

END
1-87
DTIC